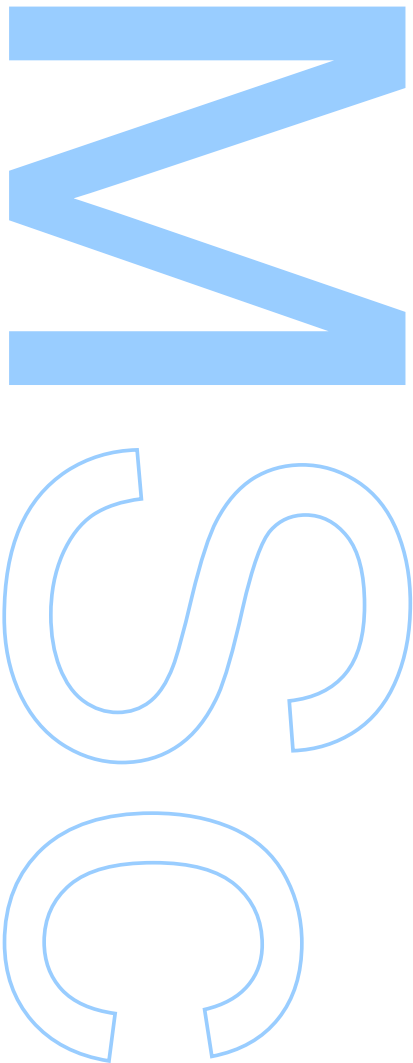
Threshold Generalization of a Contagion Model on Complex Networks

Guilherme Machado
Mestrado em Física (Especialização em Teórica)
Departamento de Física e Astronomia
2019

Orientador
Gareth Baxter, Investigador Principal, Departamento de Física, Universidade Aveiro

Coorientador
Alexander Goltsev, Investigador Auxiliar, Departamento de Física, Universidade Aveiro



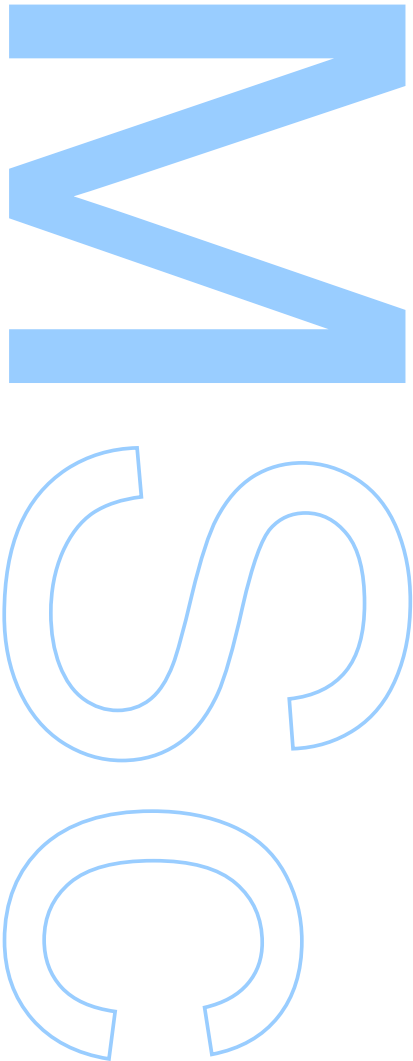
U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, _____/_____/_____



UNIVERSIDADE DO PORTO

MASTER THESIS

Threshold Generalization of a Contagion Model on Complex Networks

Author:
Guilherme Machado

Supervisor:
Gareth Baxter

Co-supervisor:
Alexander Goltsev

A thesis submitted in fulfilment of the requirements
for the degree of Mestrado em Física (Especialização em Teórica)

at the

Faculdade Ciências
Departamento de Física e Astronomia

January 2020

" A verdadeira natureza dos homens vê-se nos momentos de desespero. "

Amadeu Henrique

Agradecimentos

Firstly, I would like to note that I deeply enjoyed studying network science, it is undoubtedly an extraordinary field. For this, I would like to start to thank to Professor Baxter, with whom most of my work was developed, not only for sharing his deep understanding of network science, but also for being always accessible, extraordinarily helpful and putting up with me over the last year. I would also like to thank Professor Goltsev, responsible for showing me this amazing field of research in the first place, and giving me the opportunity to work with him.

Gostaria também de agradecer aos Professores que me inspiraram, e moldaram a primeira parte do meu caminho. Em particular ao Professor Armando, às Professoras Manuela de Física e Química, ao Professor Vitor Raposo, e à Professora Sandra Albuquerque.

Numa nota mais pessoal, gostaria de agradecer à minha mãe Cristina, companheira da "insónia", pela ultra-disponibilidade e prestabilidade, e claro, grande paciência. À minha avó Regina, por ser a pessoa mais altruísta, com quem posso sempre contar, e por ter sempre aquele "feijãozinho" à mão de semear. Ao meu pai Henrique, pelo grande exemplo que é, e pela sabedoria que me imprime. À minha irmã Madalena, pelo sorriso inalterável e aura que contagia. Ao meu irmão Sebastião, por ser responsável por 60% do que eu sou, e por estar sempre pronto a "levar uma sova" a jogar à bola. Ao meu grande amigo Manso, por provar a tartaruga na história da lebre. À Kika, por ser o meu amigo de mais longa data. À Paula, ao Guedes, ao Rafa, e ao Chico, por serem "entes" que me marcaram profundamente, e me mudaram. Aos Luís Oliveira[2], e ao Paulo, por serem os meus parceiros "no crime". Ao Filipe, ao Rodrigo, ao Nuno e ao Mitacla, os "reis da biblioteca", por saberem mais que os próprios livros, e o partilharem. Aos "Argue-nots", por serem os meus meus compinchas na discordia, mas sempre com um bom prato "nos beiços". À "Irmandade", pelos bons momentos passados. E por fim, ao melhor dos dois mundos, à "Voz da Razão" e minha companheira, a Alice! Por todas estas coisas, e por outras ainda que desconheço já que nunca me pára de surpreender, nem de me desafiar.

A todos estes, e aos restantes compadres e comadres que no meu caminho foram deixando pegadas, partilhando grandes histórias, e pequenos momentos, um obrigado, e um grande abraço!

UNIVERSIDADE DO PORTO

Abstract

Faculdade Ciências
Departamento de Física e Astronomia

Mestrado em Física (Especialização em Teórica)

Threshold Generalization of a Contagion Model on Complex Networks

by Guilherme Machado

In this thesis we propose the SIR-Bootstrap Model, where we combine the SIR Model and the Bootstrap Percolation Model on complex networks. The SIR-Bootstrap Model requires, for the transmission of the contagion agent, not only that susceptible members of the population make, probabilistically, contacts with infected ones, but also imposes a threshold of k simultaneous contacts. This makes the SIR-Bootstrap Model have properties of the two models it is based on, such as second order and hybrid phase transitions, while being more general thus having more applications.

In the limit where $k = 1$, the SIR-Bootstrap Model meets the standard SIR Model. In this limit we proved that the classical way of measuring the Giant Connected Component made exclusively of Recovered nodes is inaccurate, with the exception that the fraction of initially infected nodes, f, goes to zero. As a solution we propose the G-GCC, which we believe measures the RGCC consistently, and captures its dependence on f. We present the analytical framework to describe it, as well as numerical results that support our work.

In the limit where $\alpha = 0$ and $\beta \neq 0$, the SIR-Bootstrap Model meets the Bootstrap Percolation Model. For this limit we describe the known theory about the Bootstrap Percolation Model, as well as applications.

While we did not manage to formulate a complete theory for the description of the SIR-Bootstrap Model, we developed numerical methods for the calculation of the critical points of the second order and hybrid phase transitions, as well as an algorithm to detect the end of hybrid phase transitions. The convergence of our results with the theory in the known limits, motivates us to believe in their global validity.

UNIVERSIDADE DO PORTO

Resumo

Faculdade Ciências
Departamento de Física e Astronomia

Mestre em Ciência

Generalização do Limiar de um Modelo de Contágio em Redes Complexas

por Guilherme Machado

Nesta tese é proposto o Modelo SIR-Bootstrap, onde juntámos o Modelo SIR e o Modelo Bootstrap Percolation em redes complexas. O modelo SIR-Bootstrap requer, para a transmissão do agente de contágio, não só que os elementos susceptíveis da população façam, probabilisticamente, contactos com elementos infetados, mas também requer um limiar de k contactos simultâneos. Isto leva a que o Modelo SIR-Bootstrap tenha propriedades dos dois modelos em que é baseado, tais como transições de fase de segunda ordem e híbridas, e ao mesmo tempo sendo mais geral e tendo mais aplicações.

No limite em que $k = 1$, o Modelo SIR-Bootstrap é equivalente ao Modelo SIR. Neste limite, provámos que a maneira clássica de medir a Componente Gigante feita somente de vértices Recuperados é inválida, com a exceção do limite em que a fração de vértices inicialmente infetados, f, tende para zero. Como solução propomos a G-GCC, que acreditamos medir a RGCC de uma forma consistente, e que inclui a sua dependência em f. Apresentamos as ferramentas matemáticas para descrever esta Componente, tal como resultados numéricos que a suportam.

No limite em que $\alpha = 0$ e $\beta \neq 1$, o Modelo SIR-Bootstrap é equivalente ao Modelo Bootstrap Percolation. Neste limite, descrevemos a teoria conhecida sobre o Modelo Bootstrap Percolation, tal como as suas aplicações.

Apesar de não termos formulado uma teoria completa do Modelo SIR-Bootstrap, desenvolvemos métodos numéricos para o cálculo de pontos críticos de transições de fase de segunda ordem e híbridas, tendo também construído um algoritmo para detetar o fim de transições de fase híbridas. A convergência dos nossos resultados com a teoria, nos limites conhecidos, leva-nos a acreditar na sua validade.

Contents

List of Figures

List of Tables

Abbreviations

SI	**S**usceptible **I**nfected
SIR	**S**usceptible **I**nfected **R**ecovered
SIS	**S**usceptible **I**nfected **S**usceptible
ON	**O**riginal **N**etwork
PN	**P**ercolated **N**etwork
PRN	**P**ercolated **R**ecovered **N**etwork
GCC	**G**iant **C**onnected **C**omponent
PGCC	**P**ercolated **G**iant **C**onnected **C**omponent
RGCC	**R**ecovered **G**iant **C**onnected **C**omponent
RBCC	**R**ecovered **B**iggest **C**onnected **C**omponent
CRGCC	**C**lassic **R**ecovered **G**iant **C**onnected **C**omponent
G-GCC	**G**uilherme/Gareth **G**iant **C**onnected **C**omponent
BGCC	**B**ootstrap **G**iant **C**onnected **C**omponent

Chapter 1

Introduction

1.1 Problem

This study proposes a model for contagion in complex networks, which is a generalization of both the Susceptible-Infected-Recovered (SIR) and the Bootstrap Percolation models.

Complex networks is an extraordinarily interesting and extensive field of research. It has applications from the study of the brain [1], to the structure of the Internet [2], from the study of stability of supply networks, such as electricity [3], to economy, and the study of how companies interact with each other [4]. In this work we studied contagion processes in complex networks.

The field of contagion processes is itself extremely wide and has numerous applications. For starters, and perhaps the most important, is the mapping of how diseases spread in a population. Throughout history we can observe epidemics such as The Great Plague, or more recently the Ebola virus, that had or could potentially have had an extreme impact on public health and life. By understanding the mechanisms by which diseases propagate, one can find strategies to fight their spread and even extinguish them (as happened for Smallpox, [5]). Secondly, these models do not limit themselves to diseases, for phenomena such as the propagation of viruses on computer networks, the spread of trends (such as rumors, or fads) through social networks, or adoption of technology, can all be described by similar methods to those in disease spreading [6].

There are a lot of models for contagion, focusing not only on the way the population gets infected (on an individual level), but also on how the population globally interacts.

The simplest of these models are Compartmental Models [7]. These models divide the population in compartments, each containing the individuals in a certain state, in relation to the disease. The most common compartments are: Susceptible, that contains the individuals that are not infected by the disease, but can be in the future; Infected, containing the members of

the population that have contracted the infection already, and can now contaminate others; and Recovered (or Removed), that includes the population that has been infected, but that has recovered with immunity to the disease (or died). These models also define the parameters of the disease, such as infection rates, or recovery rates, and assume that every person is equal and interacts in the same way. The system evolves by letting everyone interact with everyone, making each person possibly change compartment. In this kind of models the disease can evolve in the following ways, depending on its parameters and on the specific model: it can die out quickly; result in an epidemic (affecting a large number of individuals in the population); or reach an endemic state (a state in which the disease survives by having always someone infected in the population). Surprisingly, these different end states can be understood through phase transitions, with critical point related to the disease's parameters. We will go in more detail over different types of compartmental models in the next section, section 1.2. With better understanding of social interactions (and better mapping of them with recent technology), compartmental models have been generalized to networks. This allowed the construction of more complex and realistic models, where assumptions such as that everyone in the population meets everyone (which we know not to be true), and not attributing individual characteristics to each node, can be rectified. In networks, people are represented by nodes, and their relationships by edges. Each node is also attributed to be in a state (compartment), and over time interacts with its neighbours (the nodes it has edges to), making it possibly change state (for example, when an infected node transmits the disease to a susceptible one). We will explore the application of the SIR Model on networks in chapter 2.

Another type of contagion models are Threshold Models, such as the Bootstrap Percolation Model. Here nodes become active (or infected) deterministically after a number of exposures to the contagion agent, rather than obtaining the infection probabilistically when making contact with an infecting individual (such as in the SIR model). Models like these serve to describe phenomena such as the activation of brain cells [8], or the spreading of rumors/information in social networks [9], where the number of contacts is the dominant factor in the spreading of the contagion agent. We will explore more of the Bootstrap Percolation Model in section 1.3, and in great detail in chapter 3.

In this thesis we propose the SIR-Bootstrap Model, where we combine the SIR Model with the Bootstrap Percolation Model. Our model can be seen as either adding the idea of a threshold to the SIR Model, or adding the idea of probable infection and recovery to the Bootstrap Percolation Model. We offer probabilities to each infected node to transmit its disease, and also to recover (as in the SIR Model), however, we require a susceptible node to be exposed a number of times, k, to the contagion agent, to become infected. This way, the SIR-Bootstrap Model considers the relevance of individual interactions, and also has under consideration

the importance of the number of exposures in the infection (activation) of a node. For this we believe the SIR-Bootstrap Model to be a more general model and to have more applications, since, not only, can it describe the same systems as the models it is based on, but also it can applied to systems that present both these properties. We present in detail the SIR-Bootstrap Model in section 1.4

In the next few sections of this chapter, we will present a brief review on three contagion models we based the SIR-Bootstrap model in: Compartmental Models and its extensions to networks, in section 1.2; and the Bootstrap Model, in section 1.3; and finally describe in detail the SIR-Bootstrap Model, in section 1.4.

In chapter 2, we will describe in detail the limit $k = 1$ of the SIR-Bootstrap Model, where it meets the standard SIR Model on networks. At first, we present a short review on Random Networks, where we analyze some important properties of Erdős-Rényi graphs, in section 2.1. Then we present the usual approach to this problem, and describe how quantities such as a Giant Connected Component made entirely of recovered nodes are calculated, in section 2.2 and 2.2.1 . Finally, in section 2.3, we prove that this approach is not valid for all values of f (as we accept in the SIR-Bootstrap Model), and suggest the creation of a new Giant Component, the G-GCC. We present the analytical framework to describe it, as well as proving that the G-GCC meets the quantities calculated using the standard approach to this problem, only in the limit $f \to 0$. In section 2.4, we present and discuss the results for this limit.

In chapter 3, we describe the limit $\alpha \to 0$ and $\beta \neq 0$ of the SIR-Bootstrap Model, where it meets the Bootstrap Percolation Model. We present the known theory of this model where we are able to calculate quantities such as the Recovered fraction of the network, as well as a Giant Component made entirely out of recovered nodes, and analyze the two types of phase transitions that appear: second order phase transitions, and hybrid phase transitions.

In chapter 4, we describe how the SIR-Bootstrap Model was simulated, as well as numerical methods that we developed for the calculation of critical points of second order and hybrid phase transitions; and an algorithm to detect the end of hybrid transitions.

In chapter 5, we present and discuss the results for the cases of $k = 2$ and $k = 3$ of the SIR-Bootstrap Model.

Finally, in chapter 6, we present our conclusions in the study of the SIR-Bootstrap Model, as well as our limitations, and finally, we propose future development and applications for it.

1.2 Compartmental Models

Compartmental models are the simplest type of contagion models. These models were first introduced by McKendrick and Kermack, in 1927 [10], and serve as a starting point in most

of the study of disease spreading.

As mentioned before these models divide the population in compartments, each containing the individuals in a certain state, in relation to the disease. Another characteristic of these models is that they are considered fully-mixed models, i.e. the system evolves by letting every member of the population interact with every other, making each individual possibly change compartment. Also, every individual is considered equal, so no individual characteristics are included, such as age, that could possibly influence the rate of recovery of a person, or personal relations, that could possibly change the probability at which one gets infected.

We now present the simplest and most famous of these models.

1.2.1 The Susceptible – Infected Model

The simplest compartmental model is the Susceptible – Infected Model (SI Model) [11]. In this model, the population is divided into only two compartments: Susceptible, and Infected, having the definitions presented before.

In this model, no recovery is possible, as such, a member of the population can only become infected, resulting in the following order of events:

$$S \to I$$

Where naturally, everyone in the final state of the disease will end up infected.

The number of susceptible individuals is defined by S, and the number of infected individuals by I, making a total of N individuals in the population.

Originally, and in the cases we present, a fixed size of the population, N, was assumed [10], implying that the lifetime of the disease was much shorter than the change in the population size, and also much shorter than the lifetime of the individuals. Furthermore, it is considered that, at start, the number of individuals who are in the Infected compartment, I, is very small in relation with the size of the population $i_0 = \frac{I_0}{N} << 1$. This makes

$$I(t) + S(t) = N \tag{1.1}$$

$$\Leftrightarrow i(t) + s(t) = 1. \tag{1.2}$$

Where $s(t) = \frac{S(t)}{N}$ and $i = \frac{I(t)}{N}$ are the fractions of the population in the susceptible and infected compartments, respectively.

Since the SI model is a fully-mixed model, it defines that at each contact, a susceptible individual will have a probability i of meeting an infected one. The model also defines β, the rate at which, each individual, makes contacts that are successful (contacts in which, if a susceptible individual met an infected one, the infection was transmitted). So, the model proposes that the rate at which all the susceptible individuals S, make contact with infected individuals, i, with success, becoming infected, is given by βiS. Making the dynamic equation of this system, in a continuous timeframe, to be:

$$\frac{di}{dt} = \beta si \tag{1.3}$$

$$\frac{ds}{dt} = -\beta si. \tag{1.4}$$

Notice that the population that becomes infected, is removed from the Susceptible compartment, and included in the Infected compartment.

One can notice that they are both first order coupled differential equations, and since we have the dependencies of i on s, we can solve them in a straightforward way. Let's start by solving the differential equation for $i(t)$. We will begin by writing it solely an function of i:

$$\frac{di}{dt} = \beta is = \beta i(1-i). \tag{1.5}$$

Now, we can observe that i is the only variable changing in time, that is, this model assumes that β is also fixed. We then obtain:

$$\frac{di}{\beta i(1-i)} = dt. \tag{1.6}$$

Stating that at $t = 0$ the disease appeared with a fraction of $i(0) = i_0$ of the population infected (the initial seed), we get

$$\int_{i_0}^{i(t)} \frac{di'}{\beta i'(1-i')} = \int_0^t dt' \tag{1.7}$$

and integrating

$$\frac{log(i') - log(1-i')}{\beta}\Bigg|_{i_0}^{i(t)} = t'\Bigg|_0^t \tag{1.8}$$

which we can write in the form:

$$i(t) = \frac{i_0 e^{\beta t}}{1 - i_0 + i_0 e^{\beta t}}. \tag{1.9}$$

Figure 1.1, shows the evolution of each compartment with time, for a clearer observation of the evolution of the model.

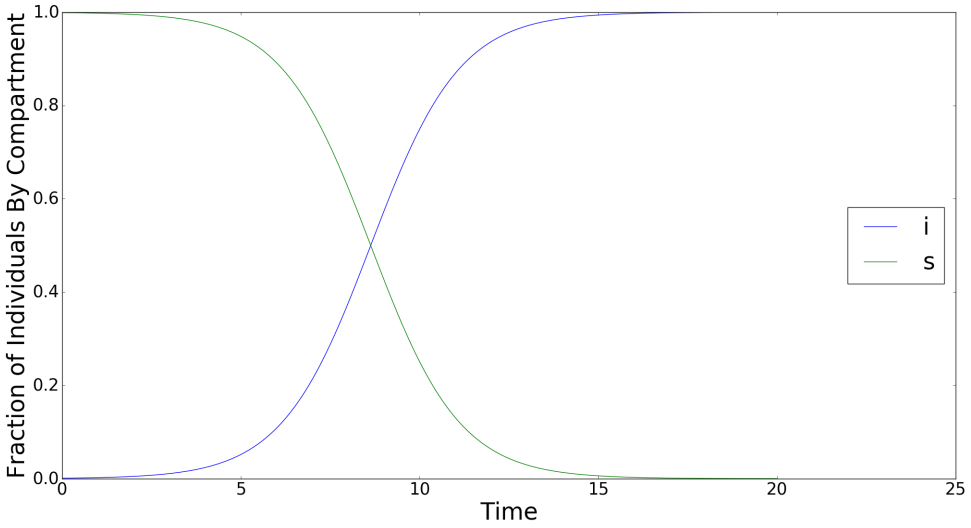


FIGURE 1.1: Evolution of each compartment with time for $\alpha = 0.4$ and $\beta = 0.8$ and $i_0 = 0.001$. The blue line shows the increase to one in the number of infected individuals, and the red line shows the decrease to zero of the susceptible ones.

Checking the limits of this model one can notice that at $t = 0, i(0) = i_0$ and that at $t = \infty, i(\infty) = 1$, meaning that in the end the entire population ends up infected, as our intuition told us. This model is an interesting start off point, but not very realistic for most diseases, since we know that not everyone ends up infected of every disease, for life. However, it can describe to some extent diseases such as herpes, AIDS, and other phenomena that upon contagion one never recovers. An example of this is the adoption of a new technology by the general population, such as happened with the spreading of the VCR, television, flush toilet [11] .

1.2.2 The Susceptible-Infected-Susceptible Model

The Susceptible-Infected-Susceptible Model, or SIS model, is an extension of the SI Model, adding the possibility of an individual to recover after being infected, becoming susceptible again. This model serves to describe diseases in which one does not gain immunity after recovery, such as gonorrhea, and diseases in which mutations make immunity ineffective, such as the flu (if we define the flu as the sum of all influenza types and subtypes; this is one of the reasons that flu vaccines are not 100% effective, and have to be changed every year, as they only contain the most common viruses for that year, and not all in absolute; if we look at an yearly timescale, the viruses the previous year, that one might have gotten immune to, have probably mutated and one's immune system might not recognize them again [12]).

The order of events of this model is:

$$S \to I \to S$$

This model introduces α, the probability that an individual recovers per unit time, becoming susceptible again. To obtain the mathematical description for this model, we would simply have to subtract the term αi from equation 1.3 and add it to equation 1.4. Making each of the infected individuals, i, recover at a rate α, leaving the Infected compartment and entering the Susceptible one again. We obtain:

$$\frac{di}{dt} = \beta si - \alpha i \tag{1.10}$$

$$\frac{ds}{dt} = -\beta si + \alpha i \tag{1.11}$$

Which again, using the relation $1 = i + s$, we can solve:

$$\frac{di}{dt} = \beta(1-i)i - \alpha i \tag{1.12}$$

$$\Leftrightarrow \frac{di}{\beta(1-i)i - \alpha i} = dt \tag{1.13}$$

$$\Leftrightarrow \frac{di}{i[(\beta - \alpha) - \beta i]} = dt \tag{1.14}$$

and integrating:

$$\int_{i_0}^{i(t)} \frac{di}{i[(\beta-\alpha)-\beta i]} = \int_0^t dt' \tag{1.15}$$

$$\Leftrightarrow \frac{log(i') - log(\beta-\alpha-\beta i')}{\beta-\alpha}\Bigg|_{i_0}^{i(t)} = t'\Big|_0^t \tag{1.16}$$

$$\Leftrightarrow log\left(\frac{i'}{\beta-\alpha-\beta i}\right)\Bigg|_{i_0}^{i(t)} = (\beta-\alpha)t \tag{1.17}$$

$$\Leftrightarrow log\left(\frac{i(t)}{\beta-\alpha-\beta i(t)}\frac{\beta-\alpha-\beta i_0}{i_0}\right) = (\beta-\alpha)t \tag{1.18}$$

$$\Leftrightarrow \frac{i(t)}{\beta-\alpha-\beta i(t)} = \frac{i_0}{\beta-\alpha-\beta i_0}e^{(\beta-\alpha)t} \tag{1.19}$$

Isolating $i(t)$:

$$i(t) = \frac{(\beta-\alpha)i_0}{\beta i_0 + (\beta-\alpha-\beta i_0)e^{-(\beta-\alpha)t}} \tag{1.20}$$

thus obtaining the evolution of the infected population over time. Analyzing this last equation, Equation (1.20), we can see that when $t \to \infty$ the system can converge to one of three states, depending on the relationship between α and β. (Note that α,$\beta > 0$)

If $\beta > \alpha$:

$$i_\infty = i(\infty) = \frac{(\beta-\alpha)i_0}{\beta i_0} = \frac{(\beta-\alpha)}{\beta} = 1 - \frac{\alpha}{\beta} \tag{1.21}$$

Figure 1.2 plots the evolution of each compartment with time for $\beta > \alpha$.

We can observe that if $\beta > \alpha$, meaning the infection rate being bigger than the recovery rate, we find an endemic state: a state in which the disease is circulating through the population maintaining always a fraction of infected individuals (even for $N \to \infty$). After the initial outbreak of the disease, the number of infected individuals increases, until it stabilizes at the equilibrium value of i_∞.

For $\beta < \alpha$:

$$i_\infty = \frac{(\beta-\alpha)i_0}{\infty} = 0 \tag{1.22}$$

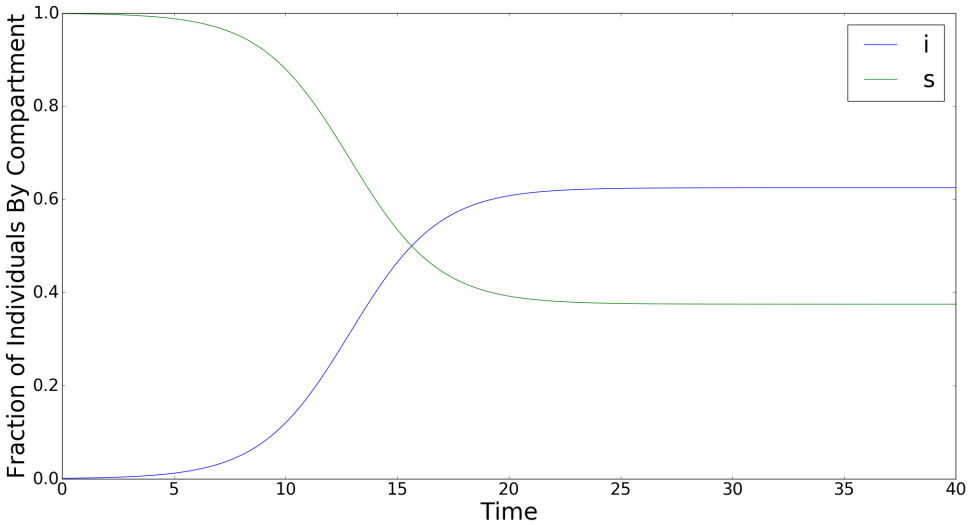


FIGURE 1.2: Evolution of each compartment with time, for $\alpha = 0.3 < \beta = 0.8$, and $i_0 = 0.001$. The infected compartment, represented by the blue line, increases from nearly zero, until it reaches an endemic state at $i_\infty = 1 - \frac{\alpha}{\beta} = 0.625$. Consequently the susceptible compartment, the red line, decreases with time until reaching $s_\infty = 0.375$.

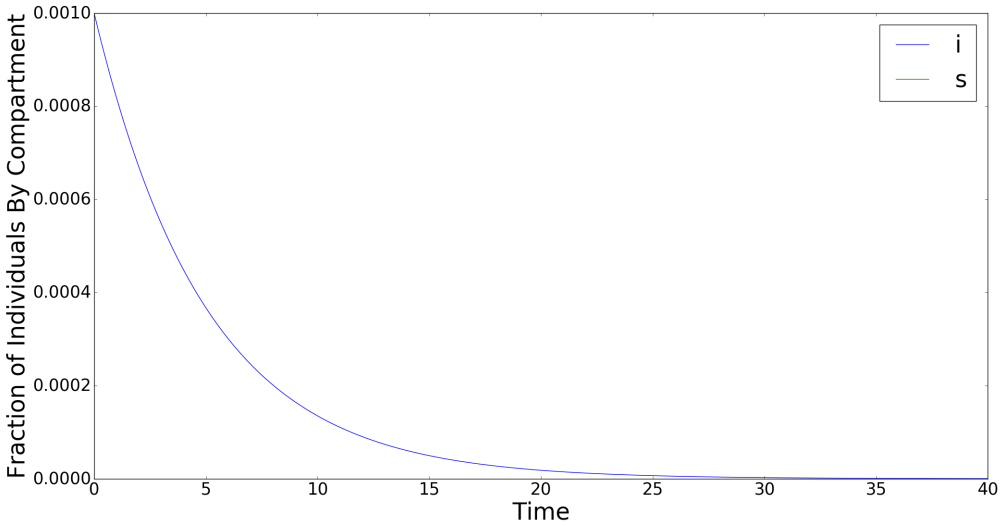


FIGURE 1.3: Evolution of the infected compartment with time, for $\alpha = 0.6$ and $\beta = 0.4$, and $i_0 = 0.001$. Since $\alpha > \beta$ we can see that the disease dies out with time quickly.

So for a recovery rate bigger than an infection rate, the disease dies out with time, as shown in Figure 1.3.

For $\beta = \alpha$ Equation (1.10) becomes:

$$\frac{di}{dt} = \beta si - \alpha i = \alpha si - \alpha i = \alpha((1-i)i - i) = -\alpha i^2 \tag{1.23}$$

Which can be solved in the following way:

$$\frac{di}{-\alpha i^2} = dt \tag{1.24}$$

$$\Leftrightarrow \int_{i_0}^{i(t)} \frac{di}{-\alpha i^2} = \int_0^t dt' \tag{1.25}$$

$$\Leftrightarrow \frac{1}{i'}\Big|_{i_0}^{i(t)} = \alpha t \tag{1.26}$$

$$\Leftrightarrow \frac{1}{i(t)} - \frac{1}{i_0} = \alpha t \tag{1.27}$$

$$\Leftrightarrow i(t) = \frac{1}{\alpha t + \frac{1}{i_0}} = \frac{i_0}{1 + i_0 \alpha t} \tag{1.28}$$

$$\Longrightarrow i_\infty \to 0 \tag{1.29}$$

As such, we can observe that the disease dies out, as in the case of $\beta < \alpha$, except that instead of $i(t)$ decreasing exponentially, for large t it decreases approximately as $\frac{1}{t}$.

Figure 1.4 plots the infinite state solutions of the infected individuals, i_∞, against β, for a fixed α. Which reminds us of a second order phase transition, since a continuous solution appears from zero, at the critical point $\frac{\beta}{\alpha}$.

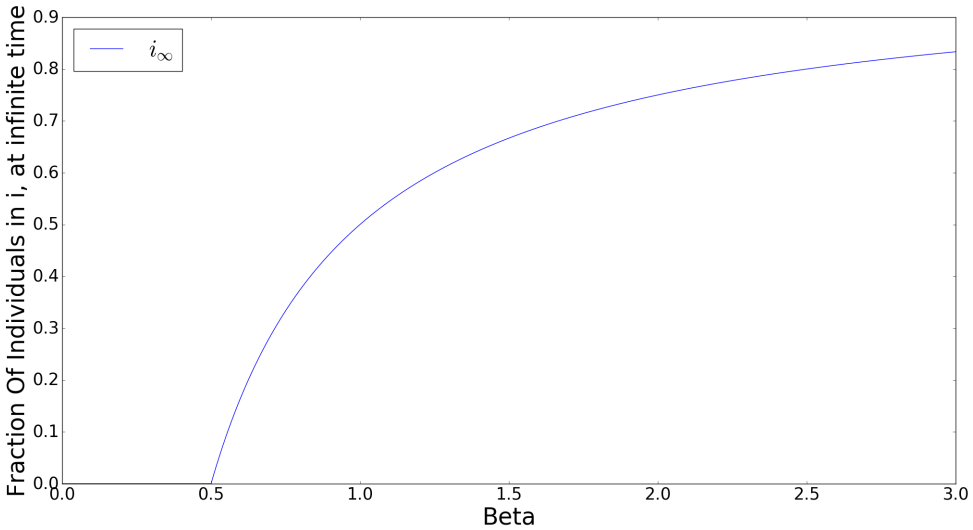


FIGURE 1.4: Evolution of the infected compartment, in the infinite time limit, with β. We fixed $\alpha = 0.5$. It is observed a critical point at $\beta = \alpha$, corresponding to the appearance of an endemic state.

1.2.3 The Susceptilbe-Infected-Recovered Model

The last compartmental model we present is the Susceptilbe-Infected-Recovered, SIR, Model. This model serves to describe diseases in which an individual recovers with immunity,

meaning that he cannot get the infection again. Most human diseases follow this type of model, such as measles, or chickenpox [7]. But this model has also been used to study phenomena such as information spread in social networks [6].

In this model an extra compartment is added, the Recovered compartment. The mathematical description of this model is close to the SIS Model, except when an individual recovers, he recovers to the Recovered compartment, and does not return to the Susceptible one (recovery with immunity).

The order of events is as follows:

$$S \to I \to R$$

As such, in the end state, we expect to get only Susceptible and Recovered individuals.

The dynamics of the system are:

$$\frac{ds}{dt} = -\beta si \tag{1.30}$$

$$\frac{di}{dt} = \beta si - \alpha i \tag{1.31}$$

$$\frac{dr}{dt} = \alpha i \tag{1.32}$$

$$s(t) + i(t) + r(t) = 1 \tag{1.33}$$

Notice that a recovery with immunity, is equivalent to death, since in both these cases an individual cannot infect, nor be infected anymore. For this consideration, we need the definitions: $s(t) = \frac{S(t)}{N_0}$, $i(t) = \frac{I(t)}{N_0}$, $R(t) = \frac{r(t)}{N_0}$, with N_0 being the original size of the population without disease deaths, or assume that the total number of deaths by disease is very small compared to the population size.

It is actually impossible to analytically solve these equations for $i(t)$ solely as a function of time. However, one can study the evolution of the size of each compartment in relation with the others, without a time scale. By dividing the equation for $i(t)$ by $s(t)$, we obtain:

$$\frac{di}{ds} = -1 + \frac{\alpha}{\beta s(t)} \tag{1.34}$$

Which we can integrate:

$$\int_{i_0}^{i(t)} i' di' = \int_{s_0}^{s(t)} -1 + \frac{\alpha}{\beta s'(t)} ds' \tag{1.35}$$

Giving:

$$i(t) = i_0 + s_0 - s(t) + \frac{\alpha}{\beta} log\left(\frac{s'(t)}{s_0}\right) \tag{1.36}$$

$$\Leftrightarrow i(t) = 1 - s(t) + \frac{\alpha}{\beta} log\left(\frac{s'(t)}{s_0}\right). \tag{1.37}$$

Even though with these equations we are only able to understand how the size of each compartment evolves in relation to the others, with the aid of computers we can simulate the full the evolution of the system, for a given α and β. We present the typical evolution of an SIR model, that we obtained using a simple Euler method, in Figure 1.5

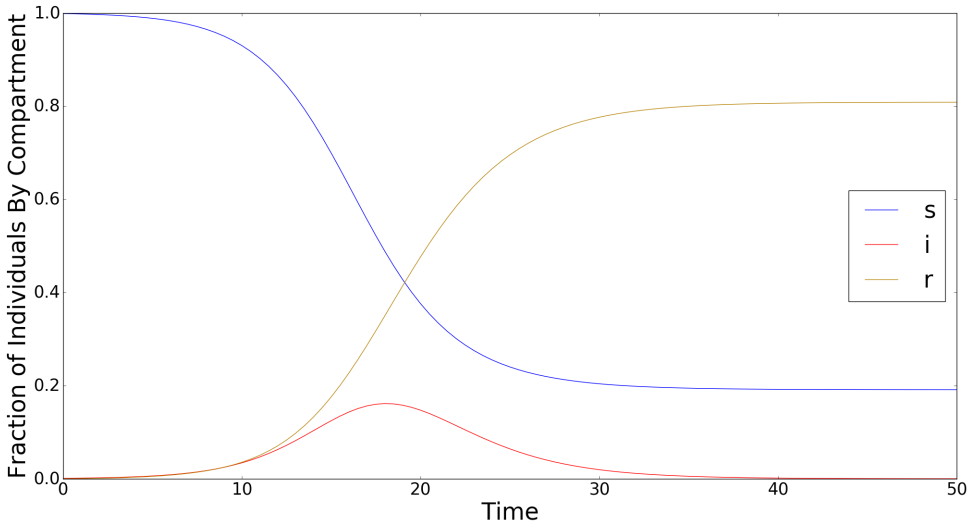


FIGURE 1.5: Evolution of each compartment with time. For the calculation of each line an Euler method was implemented. The value of the parameters used in the simulations were: $\alpha = 0.4$ and $\beta = 0.8$, and $i_0 = 0.001$. The algorithm would be stopped when the value of the infected compartment was below $\epsilon_1 = 0.0001$.

We can observe from the graphic that the number of susceptible individuals decreases with time, as some of them get infected, increasing the size of the Infected compartment. At the

same time, some infected individuals leave their compartment (by recovery or death) and are introduced in the Recovered compartment. Notice that the recovered fraction of the network at $t \to \infty$ corresponds to the total decrease in the number of susceptibles (since i_0 is very small). Just as in the SIS model, an outbreak (an increase in the initial size of the infected compartment) only exists when $\beta > \alpha$, when the infection rate is bigger than the recovery rate. In fact, the ratio of these two quantities defines $R_0 \equiv \frac{\beta}{\alpha}$, called the basic reproduction number. This is a threshold quantity that by itself tells us much about the disease, since, from these models, we conclude that the condition $R_0 > 1$ is necessary for an outbreak to occur.

1.2.4 Herd Immunity

Looking at the differential equations for the SIS and SIR models for $i(t)$

$$\frac{di}{dt} = \beta si - \alpha i = i\,(\beta s - \alpha) \tag{1.38}$$

we can notice that, at the start, $t = 0$, the only way a disease can have an outbreak is if the factor $\beta s_0 - \alpha$ is positive.

$$\beta s_0 - \alpha > 0 \tag{1.39}$$

$$\Leftrightarrow s_0 > \frac{\alpha}{\beta} = \frac{1}{R_0} \tag{1.40}$$

This allows us to write a second necessary condition for an outbreak to occur, depending only on the initial fraction of the population that is susceptible, and the basic reproduction number. Notice that, whilst we cannot do much about R_0, since it is mostly defined by the properties of the disease, if we are able to change s_0 to a value below R_0^{-1}, we could stop any infection from having an outbreak. This is where the concept of herd immunity comes from, where a whole population is secure, by having a smaller proportion of individuals who are vulnerable to a disease, then $\frac{1}{R_0}$. A way to decrease s_0, is true vaccination, since vaccinated individuals will be removed from the Susceptible compartment and introduced in the Recovered compartment, making $r_0 \neq 0$. There are a number of diseases for which, through vaccination, it was possible to greatly reduce the number of infected individuals, such as measles, rubella or polio [13]. In fact, in the case of the disease smallpox, a global eradication has been achieved, declared by the World Health Organization in 1980, with the last known natural case reported in 1977 [5]. Smallpox had a basic reproduction number

of $R_0 = 5$, requiring an immunization of 80% of the world population, for this outbreak threshold to be achieved.

1.2.5 Generalizations

Plenty of extensions have been made to the initial compartmental models, some introducing extra compartments, such as an Exposed compartment, E, attributed to individuals that have just been infected, but are incubating the disease not being able yet to infect others, making a SEIR model; others stating that recovered individuals only stay immune for so long, returning to the susceptible compartment afterwards, this would make a SIRS model; and others changing parameters such as β, giving it a dependence with time, or allowing the population size to change over time introducing birth and death rates (all these extensions can be seen in [7]). Surprisingly, even simple assumptions, such as the ones made in the original Kermack-McKendrick models, work notably well at describing outbreaks of diseases such as the Great Plague of London [7]; and the more complex models are even able to make a seasonal prediction of the population infected by some specific diseases, such as childhood diseases (for example measles) [7].

It is also possible to prove for a lot of SIR-like models that for a finite α and β, there will be a fraction of the population, s_∞, that will never be infected, such has been done in [7] for the SIR model and the SEIR models.

The detail and complexity of compartmental models can be quite deep, however, assumptions such as allowing everyone to meet everyone in the population, and not recognizing individual characteristics in the members of the populations, that are quite unrealistic, have inspired the application of these models on networks. In a network, a node would have the status of a person, and its edges the people he knows, or regularly meets. So, by changing the topology of the network, and of the individual interactions between the nodes, one can obtain much more complex, but also more realistic and versatile, models of how diseases spread.

The standard SIR model on networks attributes each node to be in one of three states (as in the compartmental models): Susceptible, Infected, and Recovered. Every infected node in the network can transmit the disease to each neighbour, each timestep, with probability β, and recover with probability α. Just like in the field of compartmental models, the application of the SIR model on networks has many variations. An interesting result is noticing that the network structure (the degree distribution) has a big influence on how the disease evolves. For example, in a SIR model on an Erdős-Rényi graph, one can observe a threshold, that separates between a disease quickly becoming extinct, or of it having an outbreak, depending on the parameters of the disease and on the network, as we will prove in section 2.2;

meanwhile in very heterogeneous networks (such as a Scale-Free network), with the exact same parameters of the disease, this threshold goes to zero [14], as such, even for a very little contagious disease, one will always observe an epidemic. A similar result can be shown for an SIS model in Scale-Free networks ([2]), where for any parameters of a disease, it will always reach an endemic state. In [15], it was shown that for scale free distributions (as is the social network), the immunization of a small fraction of the nodes of highest degree (the nodes with the most connections), can stop any outbreak from occurring, and other types of strategies in disease control are suggested; in [16], it is suggested that through a mapping of the airport network, together with the application of an SIR-like model in each node, it is possible to track the evolution of the H1N1 influenza virus in 2009, and the SARS virus in 2003; the application of the SIR model to information spread in networks has also been made in [6]. We will explore in detail the application of the standard SIR model on an Erdős-Rényi graph, in chapter 2, as it is one of the models we based the SIR-Bootstrap Model on.

1.3 Bootstrap Percolation Model

The second model we will analyze is the Bootstrap Percolation Model.

Bootstrapping is a term that traces back to the midst of the 19th century used to describe an impossible task as is "pulling one by one's own bootstraps". Over the years, this concept evolved to describe self-starting processes: processes in which a certain set of rules and initial conditions are set, from which the system evolves without any further input.

Systems like these exist from biology, to computing technology, to physics. An example of such system is the famous "Conway's Game of Life": this game consists in a lattice in which each square (cell) is set as empty (dead) or filled (alive). Over time, each cell can either die, be reborn, be kept alive or dead, based on a certain set of rules depending only on the system distribution, possibly creating dynamic patters, or reaching a stable state [17].

Here we will explore the concept of Bootstrap applied to contagion in complex networks. We define that each node can be in one of two states: active (infected) or inactive (susceptible). If an inactive node has at least k of its neighbours who are in an active state, the node will jump to the active state itself. After a node is active, it will not leave this state. A fraction of the network, f, is activated at start. So, the state of each node can evolve in the following order of events:

$$\text{Inactive} \longrightarrow \text{Active}$$

This process can be seen as a Susceptible-Infected (SI) kind of process, since each node after being infected, becomes itself infected and can infect others, never recovering. The difference

is that the nodes are not infected from each infected neighbour with a probability β, each timestep, rather they are infected deterministically once they have obtained a number, a threshold, of infected neighbours $\geq k$.

This type of model serves to describe processes in which contagion is dominated by the number of exposures (contacts) rather than each individual contact. An example of such processes is the activation of brain cells, which upon having a certain number of neighbours that are active, become active themselves sending an electric signal to their forward neighbours (it is a directed network) [1]. The application of Bootstrap Percolation in neural networks has been studied in [8], [18] and [19]. Another application is to the spreading of information in social networks (social contagion), such has been done in [9]. In [20] and [21], the Bootstrap Percolation Model is applied in the study of jamming and rigidity transitions and glassy dynamics, and in [22] to magnetic systems.

In a more theoretical understanding, the Bootstrap Percolation Model has been widely studied in regular d-dimensional lattices [23], random networks and infinite trees [24], considering constant or changeable value of k for each node [25]. And, in chapter 3, we will present the study of this process on an infinite Erdős-Rényi graph developed by Baxter *et al.* in [24], and later extended by Di Muro *et al.* in [26].

We choose this type of network for its presented advantages, shown in section 2.1, however, the theory we present applies to all networks who are locally like a tree.

1.4 The SIR-Bootstrap Model

In this thesis we present the SIR-Bootstrap Model. The SIR-Bootstrap Model proposes a contagion process which is a combination of the SIR Model with the Bootstrap Percolation Model. It provides each infected individual probabilities to infect others and recover, but at the same time only allows susceptible individuals to become infected (active) after a certain amount, k, of contacts at the same time.

The process is constructed in the following way: there is a network with N nodes, where each can be in one of three states at a given time: Susceptible, Infected or Recovered. At the start ($t = 0$), a fraction of the nodes, f, chosen uniformly at random, are set to the Infected state. These nodes are called the seeds. A susceptible node (a node in the Susceptible state) can only evolve to the infected state, and an infected node (a node in the Infected state) can only evolve to the Recovered state. Once a node is recovered (a node in the Recovered state), it can no longer leave this state. Thus the system evolves similarly to the SIR model:

$$S \longrightarrow I \longrightarrow R$$

In this model, time is discrete, and at each time, each node makes contact with its neighbours. Upon making contact, each infected node will attempt to transmit its infection, being successful in this attempt (that we define as a successful contact) with probability β, to each neighbour independently. If a susceptible node experiences k or more successful contacts, in the same timestep $t = t'$, it jumps to the Infected state. Also, at each time, each node in the Infected state has the probability to recover (to jump to the Recovered state) of α. So, as in the SIR Model, the network in the infinite time limit will be constituted solely of susceptible and recovered nodes.

This model can be seen as either a generalization of the SIR type of contagion, requiring for each node that is infected, that it has $k \geq 1$ neighbours that transmitted the contagion agent to them (not just one, as in the SIR Model); or as a generalization of the Bootstrap Percolation Model, where a recovery and an infection probability are introduced, making, not only, a susceptible (deactivated) node not deterministically infected (activated) when it has k infected (active) neighbours, but also that each infected (active) node is only contributing to the infection (activation) of its neighbours during a finite amount of time.

The SIR-Bootstrap Model is a new type of contagion process, that holds advantages of both these types of process. This model considers individual types of interaction (as does the SIR Model), but also has under consideration the importance of the number of exposures to the infection (activation) of a node. As such, not only does the SIR-Bootstrap Model describe the systems that the other two models described, but also it can be applied to systems that present both these properties, making it more general and have more applications. In the limit $k = 1$ this model becomes the standard SIR model on networks, and in the limit where $\alpha \to 0, \beta \neq 0, t \to \infty$ this model becomes the Bootstrap Percolation Model. As such, in all the examples of applications of these two models, we can also apply our SIR-Bootstrap Model, by adjusting the α, β and k parameters.

Notice that normally the SIR Model just considers the limit $f \to 0$, whilst here, due to the model's relation with the Bootstrap Model, we will allow it to be finite, for which we will have to introduce a new definition of the Recovered Giant Connected Component.

We should note that for SIR-Bootstrap Model to meet the Bootstrap Percolation Model, it is just needed that $\alpha \to 0, \beta \neq 0$, and not $\alpha \to 0, \beta = 1$. The first condition, that $\alpha \to 0$ is just noting that an infected node will never recover, thus, that it is always able to infect its neighbours, as does an active node in the Bootstrap Percolation Model. The second condition, $\beta \neq 0$ exists because since the infected nodes never recover, whatever value of β we have (except $\beta = 0$), will (eventually) result in the same infected nodes as the active nodes in the Bootstrap Percolation Model. In the Bootstrap Percolation Model the activation of a node is deterministic (guaranteed) as soon as a node has k active neighbours. In the SIR-Bootstrap Model, from the moment a susceptible node has k or more active neighbours,

it has a probability of them infecting it, each time. However, if they never recover, in the limit $t \to \infty$, there will eventually be a time in which at least k of the susceptible's infected neighbours will make successful contacts infecting it (i.e the probability that a susceptible node, having $N_I \geq k$ infected neighbours, never gets infected in the infinite time limit is: $\left[\sum_{n=0}^{k-1} \binom{N_I}{n} (1-\beta)^{N_I-n} \beta^n\right]^t = (<1)^\infty \to 0$). So the infected nodes in the infinite time limit in the SIR-Bootstrap Model, will be the same as the active nodes in the Bootstrap Model, using these parameters. Furthermore, if we set $\beta = 1$, each susceptible node, as soon as it has k infected neighbours, will be deterministically infected as well. In this limit, the SIR-Bootstrap Model evolves over time exactly as the Bootstrap Model.

An application that can be compared when described by the SIR-Bootstrap Model or by the Bootstrap model, is the spreading of rumors/ideas. In the Bootstrap Percolation Model, the spreading of a rumor corresponds to a person after hearing a rumor k times, deterministically sharing and convincing all his neighbours (the people he knows) of it. In the SIR-Bootstrap Model we consider that each time a person hears a rumor, he has a probability of believing in it (or to deem it relevant enough to share with his friends) from each source, and only if he is convinced of this rumor from k different people in the same interval of time, does he start to believe in it himself and possibly spread it. Say a person is at a dinner with his friends, the probability β corresponds to each of his infected friends to talk about the rumor in the first place, but also to him believing in each of them. Unlike the Bootstrap Model, we consider that there is a possibility of an infected person not sharing the rumor, and also that a susceptible person is not convinced by his friends; in fact, we consider even cases where a person might have agreed with some of his friends, but that the argumentation of the others was not enough to finally convince him (or make this subject relevant enough for him to share with other people). Furthermore, in the SIR-Bootstrap Model, after a person is convinced, he will only spread the rumor to his friends during a finite time interval (until he recovers), that corresponds to the interval that this rumor is new, and of relevance to him, unlike an active node in the Bootstrap Model. Other examples of this are TV series, news, fashion, technology, and so on, that when described by the Bootstrap Model, would make a person deterministically watch them, or share them upon having listened about it k times, whilst in the SIR-Bootstrap model each person is offered the possibility of not deeming it to be interesting enough to grow interest in it, to have interest and not share it with others, and even to lose interest in this agent (series, news,...) after some time.

Notice than in the way the time variable was introduced, the interval between each iteration does not have to be very small, it can be a day, a week, a month, and so on. What happens is that β and α will represent the probability of successful contact and recovery, respectively, within this time interval. So, if $\delta t = 1$ represents one hour, α will represent the probability of recovery within that hour. However, if α represents a week, α is the probability of recovery

within that week, and β is the probability that during this week, that an infected neighbours makes (at least) a successful contact via an edge. In our example of the rumors, it would represent the probability that over the course of one week that an infected person (a person that knows about the rumor) talks about a it (and generates interest) with each friend. Notice that the k successful contacts a susceptible node needs to make to get infected, do not have to be in the same instant as in the dinner example presented before, they just have to be in the same time interval considered (in this case, the whole week).

A similar type of model has been studied in [27], that differs from our model in the fact that a cumulative type of infection is considered. In this model, when a person (node) hears a rumor (or a friend talk about a series) he as a probability of believing in (liking) it (i.e. the infected node has a probability of making a successful contact), and after a susceptible person is exposed to k successful contacts, during his whole lifetime, he becomes infected. Each node is also (implicitly) offered the probability to recover. We believe that our model is more realistic since if a person heard of a rumor, or something about a series, a long time ago (say ten years ago), there is the chance that he forgot about it. So, this successful contact should not influence a person, as much as a successful contact made the day before. This is why, in our model, the successful contacts are not cumulative; for a node to become infected, it has to have made k successful contacts in the same interval of time, from t to $t+1$ (that can represent a period such as a day, a week, a month,...).

Other types of system were considered such as introducing the probability β' (instead of β), that a susceptible node jumps to the Infected state, upon having k active neighbours at a time, that is, β' applies to the susceptible node, and not to the infecting edges. In this case, each individual contact between an infected node and a susceptible one loses relevance, as what matters is if in their total, the infected individuals infected the susceptible one (in our example of the rumor, it would represent a node believing the rumor/idea from the total of the conversation at dinner, rather than from each individual interaction). There is a not linear map between β' and β, for example, a susceptible node that has exactly k infected neighbours, would have a probability of becoming infected of β' in this "global" Model, and a probability β^k in the SIR-Bootstrap Model, per unit time. However, for a higher number of infected neighbours, we would have to introduce combinatorial terms. We did not chose this type of interaction as we believe each node can only be responsible for its type of interaction with its neighbours. Moreover, β comes more naturally as a generalization of the SIR Model, where individual interactions have a high relevance.

Another thing that was considered was the introduction of a different type of recovery: τ a fixed time interval that each node would remain in the Infected state, after its infection. Figure 1.6 compares the probability of a node not recovering over time, using α or τ.

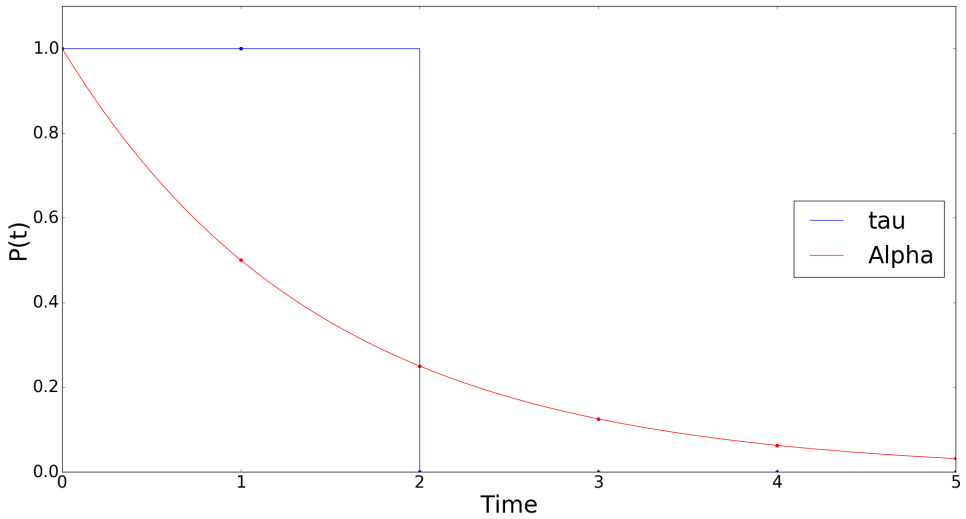


FIGURE 1.6: The evolution of the probability distribution of α and τ over time. The dots represent the probability that a node did not recover until that time, depending on whether we are using α or τ. The lines are merely illustrative, since time is discrete. The blue color represents τ, and the red one α. The values used were $\tau = 2$ and $\alpha = 0.5$

We did not opt for τ as we believe that a probabilistic recovery is more realistic than a deterministic one, however, note that the qualitative results would be the same. Also, a probabilistic recovery allows us to change α continuously, meaning we can explore more easily the whole state of spaces of α, especially at the limits $\alpha \to 1$ ($\tau = 0$), and $\alpha \to 0$ ($\tau \to \infty$, the Bootstrap limit).

Finally, the attribution of individual values of α, β and k to each node was considered, but chosen to be a work for the future after this simpler version was understood.

We should note, that from here on, everytime we refer to a curve, we mean the evolution of the Giant Component (any of them) with λ (the average number of neighbours), for a fixed α, β, f and k.

Chapter 2

Standard SIR Model on Networks and the G-GCC

In this chapter we will study the limit of the SIR-Bootstrap Model, when $k = 1$. As we have seen, in this limit the SIR-Bootstrap Model becomes the standard SIR Model on networks. However, before describing this limit in detail, we will first present some important results for networks, in particular for Erdős-Rényi graphs, that we will use in our arguments.

2.1 On Random Networks

A very famous type of network is Erdős-Rényi graph, introduced in 1959 by Paul Erdos and Alfréd Rényi [28]. Even though there are more complex model of networks, which are perhaps more realistic models for social networks, this type of network will be the starting point in our study since it has some very nice properties that make it analytically (and numerically) easy to deal with.

An Erdős-Rényi graph (ER) is constructed by defining two quantities: the number of nodes, N, in the network; and the probability, r, that an edge between two nodes exists. The graph is then constructed by asking to all possible pairs of nodes, if an edge will exist between them or not, with probability r, in an independent way. After this process is done, we have our random network constructed, with some nodes having a lot of neighbours, others having less, and even, possibly, some having none. The number of edges a node has is defined as its degree. If two nodes connect to each other via an edge, they are defined as being neighbours.

Because our construction of the edges is done in an independent way, the probability of a node having degree q is:

$$P(q) = \binom{N-1}{q}(r)^q(1-r)^{N-1-q}. \tag{2.1}$$

The node had q other nodes which he was successfully connected to, and $N-1-q$ (the remaining of them) to which he was not; all this multiplied by the possible combinations with this set. The -1 factor exists because the node cannot connect to itself.

The mean degree in an Erdős-Rényi graph is given by:

$$\langle q\rangle = \sum_{q=0}^{N-1} qP(q) = \sum_{q=0}^{N-1} q\binom{N-1}{q}(r)^q(1-r)^{N-1-q} \tag{2.2}$$

$$= \sum_{q=0}^{N-1} q\frac{(N-1)!}{q!(N-1-q)!}(r)^q(1-r)^{N-1-q} \tag{2.3}$$

$$= r(N-1)\sum_{q=1}^{N-1}\frac{(N-2)!}{(q-1)![(N-2)-(q-1)]!}(r)^{q-1}(1-r)^{(N-2)-(q-1)}. \tag{2.4}$$

Where we start the summation in $q=1$ since the term $q=0$ is zero. Thus,

$$\langle q\rangle = r(N-1)\sum_{q=1}^{N-1}\frac{(N-2)!}{(q-1)![(N-2)-(q-1)]!}(r)^{q-1}(1-r)^{(N-2)-(q-1)} \tag{2.5}$$

$$= r(N-1)\sum_{q=1}^{N-1}\binom{N-2}{q-1}(r)^{q-1}(1-r)^{(N-2)-(q-1)} \tag{2.6}$$

$$= r(N-1)\sum_{q'=0}^{N-2}\binom{N-2}{q'!}(r)^{q'}(1-r)^{(N-2)-q'}. \tag{2.7}$$

In which the summation term equals one, since we are summing over all terms of the binomial distribution. So:

$$\langle q\rangle = r(N-1) \tag{2.8}$$

Advantages of an Erdős-Rényi graph:

The first advantage of an Erdős-Rényi graph, is that $P(q)$ converges to a Poisson distribution, through a couple of useful assumptions. This quality makes random networks much easier to manipulate in analytical calculations.

Let us define $\lambda = rN' = \langle q\rangle$, with $N' = N-1$. Then

$$\binom{N'}{q} r^q (1-r)^{N'-q} = \frac{N'(N'-1)...(N'-q+1)}{q!} \left(\frac{\lambda}{N'}\right)^q \left(1-\frac{\lambda}{N'}\right)^{N'-q} \tag{2.9}$$

$$= \frac{N'^q (1-\frac{1}{N'})...(1-\frac{q-1}{N})}{q!} \left(\frac{\lambda}{N'}\right)^q \left(1-\frac{\lambda}{N'}\right)^{N'\left(1-\frac{q}{N'}\right)}, \tag{2.10}$$

assuming $N \gg 1$, and also that $N \gg \lambda = constant$:

$$P(q) \approx \frac{N'^q}{q!} \left(\frac{\lambda}{N'}\right)^q \left(1-\frac{\lambda}{N'}\right)^{N'} = \frac{\lambda^q}{q!} \left(1-\frac{\lambda}{N'}\right)^{N'} \tag{2.11}$$

$$P(q) \approx \frac{\lambda^q}{q!} e^{-\lambda}. \tag{2.12}$$

The assumption $N \gg 1$ is just saying that the network is big. The second assumption, that $N' \gg \lambda$ implies that: firstly, p is small; and secondly (and as a result), that $q \ll N'$ for any node in the distribution, since the binomial distribution rapidly decreases to zero in these conditions (i.e. all the nodes will have a number of neighbours much smaller than N).

The second advantage is that the number of finite cycles (triangles, squares,...) disappears in relation with the network's size, for fixed mean degree $\langle q \rangle$ and as $N \to \infty$ (assumptions that have been already made in the first advantage). This happens for infinite uncorrelated random networks that have a finite second moment of their degree distribution, as is the case of Erdős-Rényi graphs [29] [30] [2].

The fact that the number of finite cycles goes relatively to zero (all the triangles, squares, pentagons,...) creates the interesting characteristic of Erdős-Rényi networks that if we center ourselves around a node, almost certainly, none of its neighbours will connect to each other; and this applies to all nodes in the network! Meaning that if we look at that node's neighbours, the neighbours of the node's neighbours', will also not connect to each other; and so on; within a certain distance, that increases with the size of the network. In the limit $N \to \infty$, a randomly chosen node will seem like the root of an infinite tree, with its branches only connecting to each other (if they connect) at infinity.

A Tree is a graph in which any pair of nodes is connected by exactly one path (there are no cycles). A Forest is a graph made by several Trees not connected between themselves. A root is the node that is at the start of the tree.

In contagion this is very useful, since what happens to a node will only depend on each of its neighbour independently, as its neighbours do not interact between themselves. This also makes the disease leave a path throughout the branches of the tree, which is possible to track as we will see in future sections.

The third advantage of an Erdős-Rényi graph, is that the node's degrees are uncorrelated; i.e. as we construct the network by asking each pair of nodes if they connect in an independent way, higher degree nodes will not have higher probability of connecting to other nodes of higher degree, neither to nodes of smaller degree.

2.1.1 The Giant Connected Component

An interesting phenomena that can exist in a network, is the Giant Connected Component (GCC). A GCC can be defined as the biggest connected component in a network, whose relative size, in comparison to the size of the total (infinite) network, is different than zero. Having defined the size of a quantity (such as a network), as the number of nodes that exist in it, makes being the biggest connected component, not a sufficient condition to be a GCC. We also need the condition that: $\frac{\text{Size of GCC}}{\text{Total Size of Network}} > 0$; with the network's size being infinite. This requirement makes the GCC also have an infinite size. It is important to study the GCC as it is where "most of the action" will take place, so it will be the portion of the network that most influences the outcome, whether we are looking at contagion, percolation, or some other process.

To calculate the fraction of the network that constitutes the Giant Connected Component, one can construct the following argument:

$$S_X = \sum_{q=1}^{\infty} P(q) \sum_{l=1}^{q} \binom{q}{l} X^l (1-X)^{q-l} \tag{2.13}$$

Noticing that, a node belongs in the GCC if, and only if, at least one of its edges leads to a neighbour who is itself in the GCC; we define X as the probability that, being myself a node, I follow one of my edges, and it leads to a node who connects to the GCC via one of its neighbours who is not me.

We define "follow an edge" as choosing an edge of a specific node, and meet the node that is at the other end. This can be seen literally as drawing the network on paper, putting our finger on top of the specific node, and pass it over the chosen edge until our finger reaches the node in the other end.

Assuming we have the quantity X, we can build the total fraction of the network that belongs to the GCC, S_X, in the way that is represented by Equation (2.13). To start, this value is represented by the probability that a node has degree q (represented by $P(q)$ in the equation). Then, having degree q, we need the probability that at least one of its neighbours connects to the GCC via other neighbours (besides itself), given by $\sum_{l=1}^{q} \binom{q}{l} X^l (1-X)^{q-l}$, where all the possible cases of any of its neighbours connecting to the GCC are taken into account. And the sum of all these values will give us the total probability that a node of degree q belongs

to the GCC. Notice that the summation over l starts in $l = 1$ as in order for the node to be in the GCC, he has to have at least one neighbour who connects to it. Finally, to obtain the final value for S_X we just need to sum over all the possible values of q.

We should also take under consideration that we are actually calculating the expected value of the relative size of the GCC:

$$S_X = \frac{\sum_q^{\infty} \text{ Number of Nodes of Degree q} \times \text{Probability of a node of degree q being in the GCC}}{\text{Total Number of Nodes}}$$

but since our network is infinite, the expected value meets the actual value. Because we are in this limit, we can also see $P(q)$ as the fraction of nodes that have degree q, which we multiply by the probability that they connect to the GCC (and then sum them) to obtain S_X.

Finally, the summation over q starts in $q = 1$, as in order to be in the GCC a node has to have at least one neighbour, to connect to it. (An isolated node can never constitute a GCC by itself (even if all the nodes in the network are isolated), since as $N \to \infty$ its relative size would go to zero, breaking the definition of the GCC).

With this, in order to calculate the fraction of nodes that belong to the GCC, we just need to obtain X. Now, we will use the fact that the network is locally like a tree. In fact, this line of thought, from here henceforth, can only be applied to networks of this sort. This is because in these kinds of networks, none of the neighbours of a node will connect to each other, and none of their neighbours' as well, and so on. So, and as we said before, a randomly chosen node can be seen as the start of an (almost) infinite tree, that only connects at an infinitely long distance.

Figure 2.1 compares a network in which we can use an argument based on the independence of each branch, against one in which we cannot.

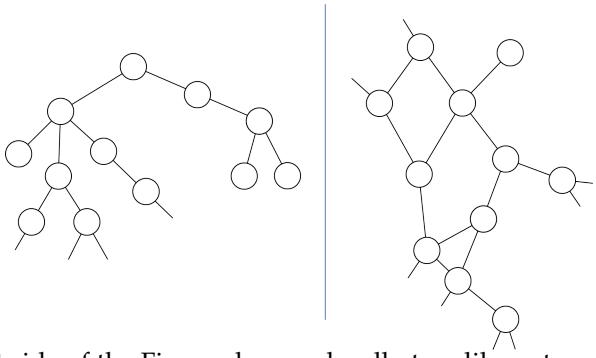


FIGURE 2.1: The left side of the Figure shows a locally tree-like network, and the right side shows a network which has a lot of finite cycles.

As a consequence, the probability that one of my neighbours connects to the GCC is independent of the probability that a different neighbour connects to it. Thus, we can treat each

branch in an independent way. If that was not the case (as happens in networks with non-vanishing Clustering Coefficient, with N), such as in the network on the right hand side of Figure 2.1, we would have take into account all the possibilities of each two nodes, that are neighbours, connecting to the GCC, via one common neighbour, making each branch not independent. Once again, since this phenomena does not happen in a relevant way in infinite random networks where the second moment is finite, we can consider the treatment of each branch in an independent way, and our argumentation here henceforth, can just be applied to networks of this sort.

With each branch being independent, we can construct X only worrying about paths of each of a node's neighbours independently. We can write X in the following self-consistent way:

$$X = \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \sum_{l=1}^{q-1} \binom{q-1}{l} X^l (1-X)^{q-1-l}. \tag{2.14}$$

For the sake of the explanation, let us say I am a node. We are defining X as being equal to the probability that I follow one of my edges and meet a node that connects to the GCC via one of its neighbours, who is not me. Now, let's assume the node I meet upon following an edge has degree q. The probability that I follow an edge and meet a node of degree q is given by : $\frac{qP(q)}{<q>}$; since, if I follow an edge it is more likely that I meet a node with more edges (higher degree), than one with less. The probability that I meet a node with degree q is the number of edges that lead to a node of degree q, divided by the total number of edges that lead to nodes. The number of edges that leads to a node of degree q is: $qNP(q)$, with $NP(q)$ being the number of nodes that exist with degree q, and q the number of edges that each of them has; hence the total number of edges that lead to nodes is: $\sum_{q=0}^{q=\infty} qNP(q) = N\langle q \rangle$. Leading to a probability of following an edge and finding a node with degree q of: $\frac{qNP(q)}{N\langle q \rangle} = \frac{qP(q)}{\langle q \rangle}$.

After meeting the node of degree q, we still need it to connect to the GCC via one of its neighbours who is not me. This is given by the factor $\sum_{l=1}^{q-1} \binom{q-1}{l} X^l (1-X)^{q-1-l}$ in Equation (2.14). This term represents that out of its $q-1$ remaining neighbours, at least one of them connects to the GCC. The -1 factor in $q-1$ appears has the edge that connects to me is left out since if we included it, it would be circular reasoning, and X would not be a sufficient condition to meet the GCC. The rest of the terms work just as before: that this node connects to l neighbours to lead to the GCC, and $q-l$ of them who are not. Of course, l can take values from 1 (the node has to connect at least once to the GCC), to $q-1$ (representing all other edges connect to the GCC, except the one that connects to me), and we have to sum over them. So far, we have calculated the probability that upon following an edge and meeting a node of degree q, that he connects to the GCC. So in order to obtain X, we just have to sum over q, covering all the possibilities of the degree of the node we meet. Finally, we should

notice that in this case q starts at $q = 2$ as any node I meet that connects to the GCC needs at least one edge to connect to me, and another to connect to the GCC.

Do notice, that the only reason we can write X this way, is because the branches are independent (i.e. the network is locally like a tree): we can write the cases for X and $1 - X$ in an independent way, for all nodes (me, my neighbour, its neighbours',...).

Now that we have X, even though in a self-consistency way, if we find a way to calculate it, we just have to plug in that value in Equation (2.13) to obtain the fraction of nodes belonging in the GCC.

A final note should be made that if the network we were taking under consideration was correlated (did not have the third advantage of Erdős-Rényi graphs), we could write these exact equations by: substituting $P(q)$ with $P(q|q')$, representing the probability that I am meeting a node of degree q, given that I am coming from a node of degree q'. This may require a different definition of X for each q'.

Before calculating the value of X (perhaps using a numerical method), there is a way to simplify its self-consistency equation. Notice that the last summation of Equation (2.14) looks like a binomial distribution. So we can use the binomial theorem

$$(x+y)^n = \sum_{l=0}^{n} \binom{n}{l} x^l (y)^{n-l} \tag{2.15}$$

to simplify it:

$$X = \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \sum_{l=1}^{q-1} \binom{q-1}{l} X^l (1-X)^{q-1-l} \tag{2.16}$$

$$X = \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} [1 - (1-X)^{q-1}]. \tag{2.17}$$

Notice that the summation over q, can start in $q = 1$, as this term is zero. We thus obtain

$$X = \sum_{q=1}^{\infty} \frac{qP(q)}{< q >} [1 - (1-X)^{q-1}], \tag{2.18}$$

and writing explicitly $P(q)$, assuming we are working in a Erdős-Rényi , we get

$$X = \sum_{q=1}^{\infty} \frac{qe^{-\lambda}\lambda^q}{< q > q!} [1 - (1-X)^{q-1}]. \tag{2.19}$$

Noticing also that $\lambda = \langle q \rangle$:

$$X = \sum_{q=1}^{\infty} \frac{e^{-\lambda}\lambda^{q-1}}{(q-1)!}[1-(1-X)^{q-1}] \tag{2.20}$$

$$\Leftrightarrow X = \sum_{q=1}^{\infty} \frac{e^{-\lambda}\lambda^{q-1}}{(q-1)!} - \sum_{q=1}^{\infty} \frac{e^{-\lambda}\lambda^{q-1}}{(q-1)!}(1-X)^{q-1}. \tag{2.21}$$

Making the change of variable $q' = q - 1$, we obtain

$$X = \sum_{q'=0}^{\infty} \frac{e^{-\lambda}\lambda^{q'}}{q'!} - \sum_{q'=0}^{\infty} \frac{e^{-\lambda}\lambda^{q'}}{q'!}(1-X)^{q'}. \tag{2.22}$$

The first term is the summation over all terms of a Poisson distribution of parameter λ and variable q', so it sums up to one: $\sum_{q'=0}^{\infty} Poisson(\lambda, q') = 1$. The second term we can also treat it in a way it transforms into a Poisson distribution, by multiplying and dividing by the factor $e^{-\lambda X}$:

$$X = 1 - e^{-\lambda X} \sum_{q'=0}^{\infty} \frac{e^{-\lambda(1-X)}[\lambda(1-X)]^{q'}}{q'!} \tag{2.23}$$

This time, we obtain the sum over all terms of a Poisson distribution of the type $Poisson\,(\lambda(1-X), q')$ which also adds up to one. This way we get:

$$X = 1 - e^{-\lambda X}. \tag{2.24}$$

The equation we reach for X is a transcendental equation, which is not analytically solvable. Luckily, since it is also a self consistency equation, we do not have to solve it, we just have to find the solution(s) that satisfy it, and then choose the one(s) that has(ve) physical meaning. This can be easily done using a computer. There are a lot of methods that exist to solve self-consistency equations, in fact, one could even write Equation (2.24) as $0 = 1 - e^{-\lambda X} - X \equiv F(X)$, and then use a root finding algorithm to calculate the zeros of $F(X)$. The way we solved this self-consistency equation was by attributing two functions $LHS(X)$ and $RHS(X)$ to the left hand side, and the right hand side of Equation (2.24), respectively. Then the solution for X would be the value of the intersection point of these two functions. In order to do this, we started by setting an initial value for X ($LHS_0 = X_0$), and then calculated $RHS(X)$ at that initial value of X ($RHS_0 = RHS(LHS_0)$). Then, we would

attribute the new value of $RHS(X)$ to X ($LHS_1 = RHS_0$), and after, calculate $RHS(X)$ at this new value of X ($RHS_1 = RHS(LHS_1)$). We would do this until we reached the condition $|RHS(X) - LHS| < \epsilon$, with ϵ being a quality threshold set by us. By using this method, the functions would necessarily evolve towards one of the stable solutions of the self-consistency equation. Figure 2.2 shows the evolution of this method.

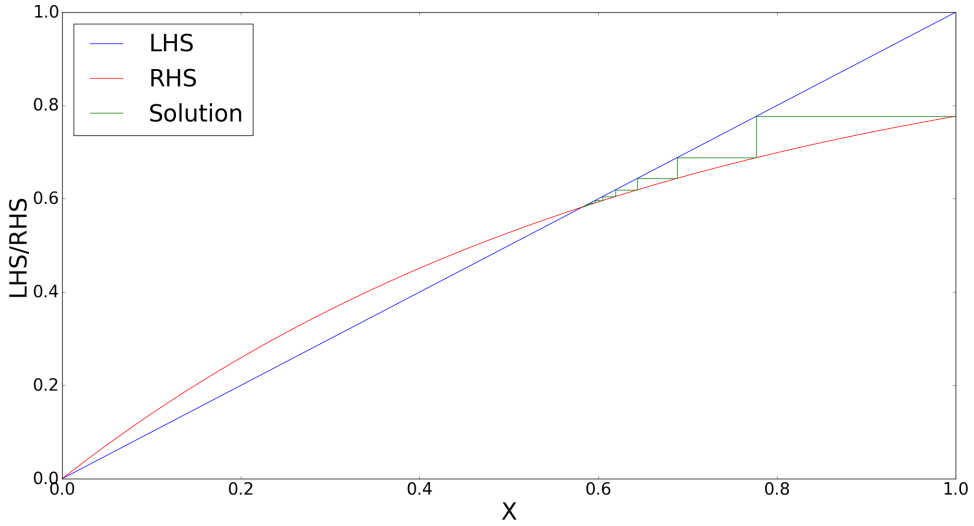


FIGURE 2.2: The green line shows the step-by-step evolution of the solution of X, using the presented algorithm. The blue line is the function $LHS(X)$, and the red line the function RHS.

We can observe that each iteration, the functions get closer and closer to the solution for X. The only way we would not reach a stable solution is if either we would start at exactly an unstable solution, or if we found a loop between two solutions. Because of these cases, and the case where the system would evolve to $X \to \infty$, we needed to be careful with the initial conditions that we set for X_0. Also this algorithm, just tells us one stable solution, so when we wanted to find the others, we plotted LHS and RHS and checked in which initial conditions we would need to start to reach them. Of course, since $LHS, RHS \in [0,1]$ this made the interval of observation of where we should start to meet each solution rather simple, as solutions out of this interval necessarily would not have a physical meaning. Lastly, one should note that we are looking for stable solutions, as unstable solutions have little physical meaning; real systems evolve to the stable ones (assuming there are no cyclical solutions).

Having found the solutions for X, one can plug it into S_X, to calculate the final fraction of the network that belongs to the GCC. Before we do that, one can simplify the equations for S_X, in a very similar manner as we did for X:

$$S_X = \sum_{q=1}^{\infty} P(q) \sum_{l=1}^{q} \binom{q}{l} X^l (1-X)^{q-l} \tag{2.25}$$

$$\Leftrightarrow S_X = \sum_{q=1}^{\infty} P(q)[1-(1-X)^q] \tag{2.26}$$

where, once again, we used the binomial theorem. Notice once more that the summation can start in $q = 0$.

$$S_X = \sum_{q=0}^{\infty} \frac{e^{-\lambda}\lambda^q}{q!}[1-(1-X)^q] \tag{2.27}$$

$$\Leftrightarrow S_X = \sum_{q=0}^{\infty} \frac{e^{-\lambda}\lambda^q}{q!} - \sum_{q=0}^{\infty} \frac{e^{-\lambda}\lambda^q}{q!}(1-X)^q \tag{2.28}$$

and once again, having obtained sums over all the possible values of a Poisson distribution:

$$\Leftrightarrow S_X = 1 - \sum_{q=0}^{\infty} \frac{e^{-\lambda}[\lambda(1-X)]^q}{q!}. \tag{2.29}$$

Which we can multiply and divide by $e^{-\lambda X}$, to obtain Poisson-like terms:

$$\Leftrightarrow S_X = 1 - e^{-\lambda X} \sum_{q=0}^{\infty} \frac{e^{-\lambda(1-X)}[\lambda(1-X)]^q}{q!} \tag{2.30}$$

$$\Leftrightarrow S_X = 1 - e^{-\lambda X}. \tag{2.31}$$

This is a very interesting result, since it tells us that in an Erdős-Rényi graph the value of $S_X = X$.

Figures 2.3 and 2.4 show the evolution of S_X and X with the mean degree.

We can observe that for small values of λ there is no giant component. And, when λ is big enough (let's call the threshold value λ_{crit}) a giant component starts to appear. Physically, this means that initially the clusters that exist in the network (that starts with only isolated nodes) are very small when compared to the network, i.e. they are finite. And as we increase λ (which can be seen has adding edges to the network) these clusters get bigger and bigger, but are still finite. Only after $\lambda > \lambda_{crit}$ these clusters start to connect to each other taking up an infinite portion of the network: the GCC. One could make an analogy to small fires in a forest (in the real world, not forest in the network sense), that grow bigger and bigger, and after reaching a certain size they start connecting to each other making a much bigger fire.

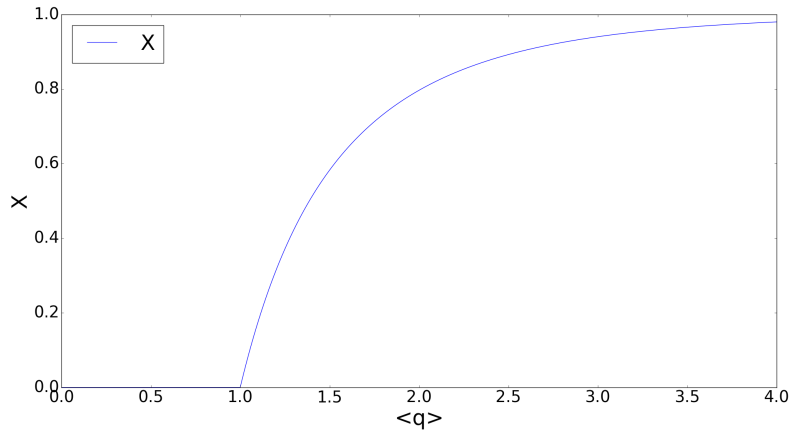


FIGURE 2.3: The figure shows the evolution of X with the mean degree.

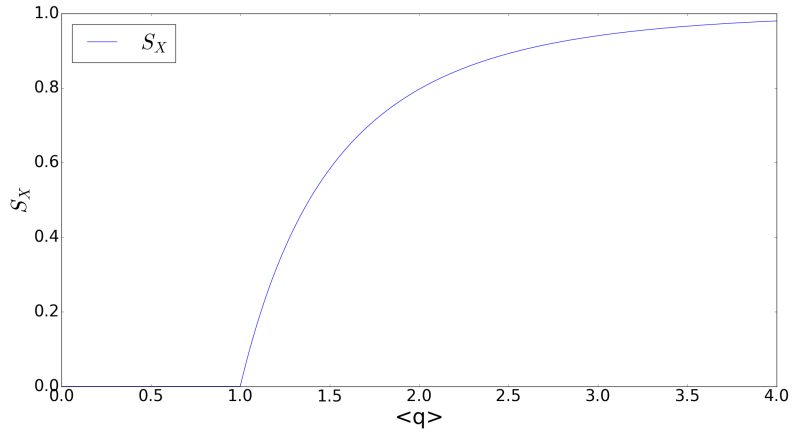


FIGURE 2.4: The figure shows the evolution of the fraction of nodes that belong to the GCC with the mean degree.

Regarding X, what happens is that the only stable solution is $X = 0$, for $\lambda < \lambda_{crit}$. And at λ_{crit} a second solution appears which is stable, and the solution for $X = 0$ becomes unstable; this corresponds to the appearance of the GCC. In fact, the appearance of the second solution is a second order phase transition, as could be intuited by the shape of S_X. For starters, the second solution appears in a continuous way from zero, and secondly, the mean cluster size (that here works like the Susceptibility, in Ising-like transitions), diverges at $\lambda = \lambda_{crit}$.

An interesting point to consider is the point where the GCC appears, the critical point in the phase transition. One can calculate it in the two following manners.

First approach:

If again we set a function to each side of the equation, we can notice that the RHS (Right Hand Side) equation is a monotonically increasing function (since $e^{-\lambda X}$ goes from one to zero monotonically, in the semi-positive plane; and the other term is just a constant). Another thing we can notice is that the first derivative of RHS, $RHS' = \lambda e^{-\lambda X}$, is monotonically decreasing, and goes to zero. This implies, that RHS grows more and more slowly as X

increases, and stabilizes in a constant for $X \to \infty$. It also implies that RHS' has a maximum at $X = 0$. This being, RHS will intersect with LHS a second time (other than at $X = 0$), if the first derivative of RHS at $X = 0$ (its maximum value) is bigger than the one for LHS. Figure 2.5 shows the possible cases of different solutions.

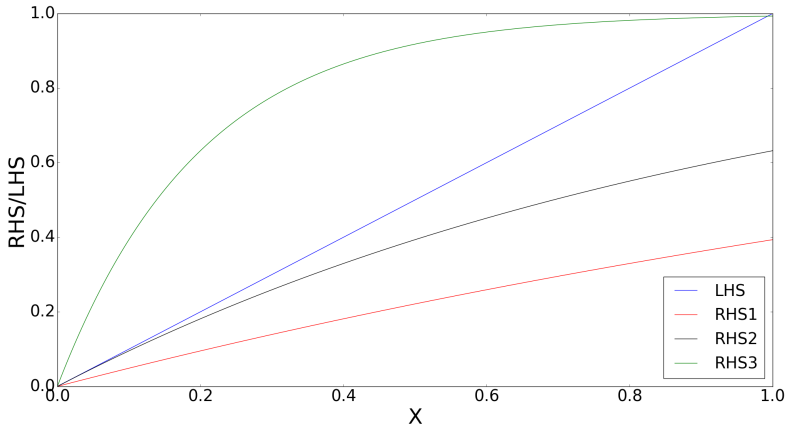


FIGURE 2.5: The figure shows the RHS function for different values of mean degree. The red line, $RHS1$, was calculated at $\langle q \rangle = 0.5$, where the solution for X is zero; the black line, $RHS2$ at $\langle q \rangle = 1$, the exact point where a solution for $X \neq 0$ starts to appear; and the green line $RHS3$ at $\langle q \rangle = 5$, where a the stable solution for X is bigger than zero.

With this, we know that:

$$LHS' = 1 \tag{2.32}$$

$$RHS'|_{X=0} = \lambda. \tag{2.33}$$

So for a second solution (and the GCC) to appear we need the condition:

$$\lambda > 1 \Leftrightarrow \lambda_{crit} = 1. \tag{2.34}$$

Second approach:

Another way to calculate the appearance of a second solution is to notice that the second solution (and the GCC) appear from zero. So we can make a Taylor expansion around $X = 0$, and calculate where this happens:

One could calculate it using the final equation for X:

$$X = 1 - e^{-\lambda X} \approx 1 - [1 - \lambda X + O(X^2)] = \lambda X \tag{2.35}$$

$$\Leftrightarrow \lambda = 1. \tag{2.36}$$

Or in a more general way, consider any distribution, $P(q)$, that respects the conditions we have imposed so far (such as finite second moment):

$$X = \sum_{q=1}^{\infty} \frac{qP(q)}{\langle q \rangle} [1 - (1 - X)^{q-1}] \tag{2.37}$$

$$\Leftrightarrow X = \sum_{q=1}^{\infty} \frac{qP(q)}{\langle q \rangle} \{1 - [1 - (q-1)X + O(X^2)]\} \tag{2.38}$$

$$\Leftrightarrow X \approx \sum_{q=1}^{\infty} \frac{qP(q)}{\langle q \rangle} (q-1)X \tag{2.39}$$

$$\Leftrightarrow 1 = \frac{1}{\langle q \rangle} \sum_{q=1}^{\infty} qP(q)(q-1). \tag{2.40}$$

We can add the zeroth term to the summation, since it is zero:

$$\Leftrightarrow 1 = \frac{1}{\langle q \rangle} \sum_{q=0}^{\infty} qP(q)(q-1). \tag{2.41}$$

Notice that the summation term is just the expected value of $\langle q(q-1) \rangle$. Which gives the condition for the appearance of the GCC of:

$$1 = \frac{\langle q(q-1) \rangle}{\langle q \rangle}, \tag{2.42}$$

which for a Poisson like distribution leads to the same result as before:

$$1 = \frac{\langle q(q-1) \rangle}{\langle q \rangle} = \frac{\langle q^2 \rangle - \langle q \rangle}{\langle q \rangle} \tag{2.43}$$

$$\Leftrightarrow 1 = \frac{\lambda + \lambda^2 - \lambda}{\lambda} = \lambda \tag{2.44}$$

where we have used that $\langle q^2 \rangle = \lambda + \lambda^2$

To sum up, in a Random Network, as we increase the mean degree, a Giant Connected Component appears at $< q >= \lambda_{crit} = 1$.

2.1.2 Percolation

A famous and interesting process to study in networks is Percolation. Percolation is the change of the global behaviour of a system, for example a lattice, in which each site can be occupied with probability p and vacant with probability $1 - p$, with the change of the parameter p. An example of this is a porous material which connects two recipients, one with

a fluid and another without. If we fill each connection of each site of porous material (the lattice) with probability p, a link between the two recipients can emerge. The appearance of this link, when p is big enough, allows the fluid to pass through, changing the total behaviour of the system.

On a different note, it is also very useful to study robustness in networks: how will a network be affected by the removal of its edges or of its nodes. Robustness can be tested either by removing edges randomly, or in a orderly manner. In fact, it is well known that Scale Free Networks (such as the Internet) are very robust to removal of random edges, when compared to Erdős-Rényi graphs [14]. This is a very interesting result since it tells us that if a percentage of the network fails randomly (by mechanical defect, power failure,...) the network will continue to work just as fine (by "works just as fine" and "is robust", we mean that quantities such as the average shortest path is little affected. So in the case of the internet, the information flow would stay roughly the same). On the other hand, if we damage the network by removing first the nodes with highest degree and then the ones with lowest, it is also known that Scale Free Networks are highly vulnerable to this kind of attack; whilst Erdős-Rényi networks are more robust [31]. This result can be used in contagion models, such as during an epidemic, since it suggests that we should vaccinate (which is equivalent to the removal of the node, and its edges) the people with highest degree first, in order to "shut down" the spreading of the disease faster. So, whether we are studying the stability of the Internet, or of a Power Grid, to disease spreading, Robustness is always useful to study.

Notice that the removal of each edge of a network with probability $1 - p$, which would be a robustness problem, can be seen as a percolation problem, in which the existent edges of the network are considered sites, that can be filled (not damaged) with probability p, or vacant (removed) with probability $(1 - p)$. From now on when we refer to percolation, we will be looking at it from this perspective, that there was a network whose edges were damaged with probability $1 - p$.

The Percolation "type" we will use in our study is the random removal of existing edges. That is, after building an Erdős-Rényi network by fixing $\langle q \rangle$ and N (the way we defined it before was by defining r and N, but one can define r through $\langle q \rangle$ since $\langle q \rangle = rN$), we will assign a probability $1 - p$ that each of the existing edges is removed. So, each originally existent edge, after we damage the network, will have a probability of still existing of p.

One can then ask whether damaging an Erdős-Rényi graph will have an impact on the formation of the GCC, and the critical point. And if so, how much will that impact be?

In fact, we can calculate these quantities in a similar manner as before. However, before we move on, we should clarify our nomenclature when referring to the Giant Connected Component before and after it was damaged. We will introduce the Percolated Giant Connected

Component (PGCC), that represents the Giant Connected Component (if it exists) after the graph was damaged. The term GCC will be used representing the Giant Connected Component in the original graph.

The way we can write the fraction of the network that belongs to the PGCC is the following:

$$S_{X_p} = \sum_{q=1}^{\infty} P(q) \sum_{l=1}^{q} \binom{q}{l} (X_p)^l (1 - X_p)^{q-l}. \tag{2.45}$$

Where X_p is defined as the probability that after damaging the graph, being myself a node, that I follow one of my edges and it leads to a node who has at least one neighbour (that is not me) who belongs to the PGCC. We can notice, that the expression for S_{X_p} is the same as for S_X by changing $X_p \to X$. This is because we can use a trick which is maintaining the same degree distribution, $P(q)$ (before the graph was percolated), for the percolated graph, but making sure that when following an edge and meet a neighbour who is in the PGCC, X_p, that the edge we are following did not get removed. Figure 2.6 shows a graph before and after it was damaged, in order to make this argument clearer.

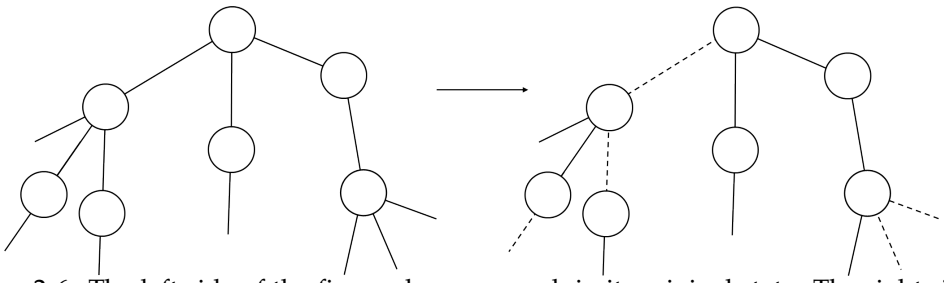


FIGURE 2.6: The left side of the figure shows a graph in its original state. The right side of the figure shows the same graph after it was damaged. The dashed lines represent the edges that were removed.

Going through the expression for S_{X_p}, we can understand that it is constructed by calculating the probability of a node having degree q (in the original graph), and that it has at least one of its edges which connects to the PGCC (this is represented by the sum over l). Then by summing over all possible values of q. All the summations here work just as before (such has starting l at $l = 1$, since each node that is in the PGCC needs to connect at least once to it,...).

With this, we just have to define this new X_p. Which we can write, as before, in a self-consistently way

$$X_p = p \sum_{q=2}^{\infty} \frac{qP(q)}{\langle q \rangle} \sum_{l=1}^{q-1} \binom{q-1}{l} X_p^l (1 - X_p)^{q-1-l}. \tag{2.46}$$

Notice this is the expression for X with an extra factor of p multiplying. This is due to the fact that, assuming I am a node, the probability that I follow an edge and meet a neighbour

who connects to the PGCC via one of its neighbours who is not me, remains the same, with the added condition that the edge I am following was not removed. In other works, we can work as if we were in the original graph (before we damaged it), and add the condition, each time we follow an edge, that that edge was not removed. Thus, X_p is the probability that I follow an edge (in the original graph), that that edge was not removed (the factor p), that I meet a node of degree q (the factor $\frac{qP(q)}{<q>}$), and that this node I meet connects to someone who belongs to the PGCC, at least once (the factor $\sum_{l=1}^{q-1} \binom{q-1}{l} X_p^l (1-X_p)^{q-1-l}$), and then of course, summing over all possible values of q to cover all possibilities. Finally, notice that the node I meet, when it connects to the PGCC, it also has to follow an edge that was not removed (it is included in X_p).

Again, we obtain a self-consistency equation for X_p, that after solving, one can introduce it in Equation (2.45), and calculate the final fraction of the network that belongs to the PGCC.

As before, we can simplify Equations (2.46) and (2.45). Notice that the equation for X_p on the right hand side is the same as the equation for X, by performing the transformation $X \to X_p$, and multiplying by p. And since the multiplication by a constant does not interfere with the derivation, we can take the final equation for X and write

$$X_p = p(1 - e^{-\lambda X_p}). \tag{2.47}$$

Regarding S_{X_p}, we can also notice that it is the same as S_X, by performing the transfomation $X \to X_p$. Thus:

$$S_{X_p} = 1 - e^{-\lambda X_p}. \tag{2.48}$$

Notice that in the limit of $p \to 1$ (no removal of nodes) we get the equations for the GCC, as we should.

We can calculate the self-consistency solutions for X_p using the same algorithm as before (by setting two functions, one to each side of the equation). As in the previous case, this equation just has one stable solution for small values of λ (for a fixed p), and then starts to have a second stable solution at a value that we will define has λ_{pcrit}. When this second solution appears, the solution for $X_p = 0$, which is always a solution, becomes unstable.

Figure 2.7 and 2.8 shows the evolution of S_{X_p} and X_p with the mean degree, respectively.

Once again, the evolution of the PGCC with the mean degree presents a second order phase transition. This transition happens with the appearance of the second solution. Notice that unlike the previous case, the existence of a p makes $S_{X_p} \neq X_p$, in fact,X_p is maximized by

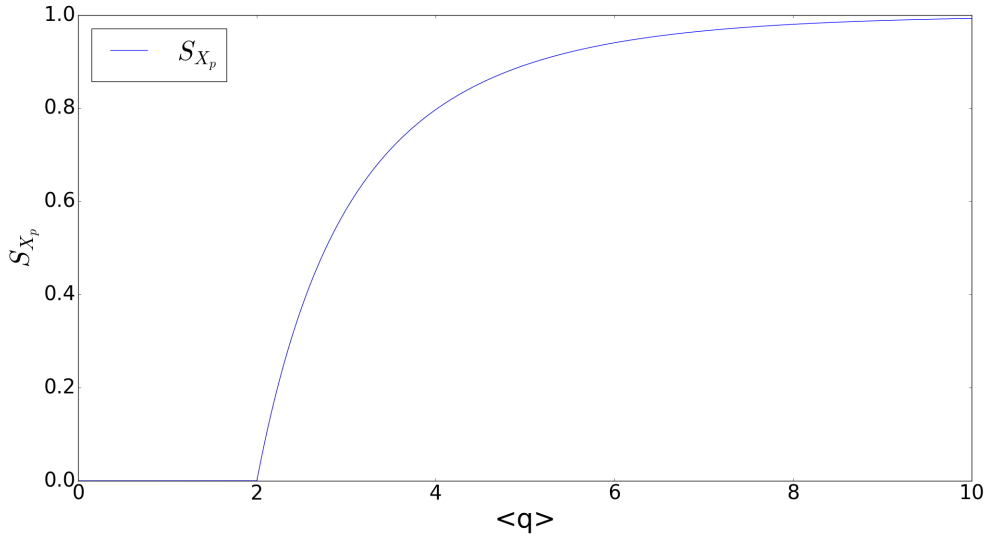


FIGURE 2.7: This figure shows the evolution of S_{X_p} with the mean degree, for the value $p = 0.5$

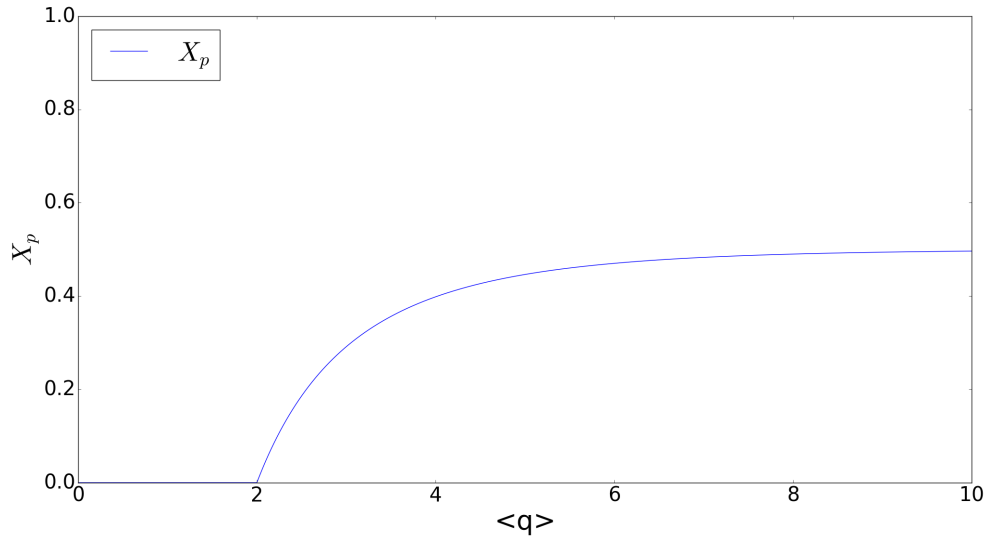


FIGURE 2.8: This figure shows the evolution of X_p with the mean degree, for the value $p = 0.5$

p. We can calculate the value of the critical point in a very similar manner has before. This time, we will just do it using the more general approach, and then specify for the Erdős-Rényi graph's case:

$$X_p = p \sum_{q=2}^{\infty} \frac{qP(q)}{\langle q \rangle} \sum_{l=1}^{q-1} \binom{q-1}{l} X_p^l (1 - X_p)^{q-1-l} \tag{2.49}$$

as in the previous case, both the PGCC and X_p appear from zero. Expanding around $X_p = 0$, and disregarding second order terms:

$$X_p = p \sum_{q=2}^{\infty} \frac{qP(q)}{\langle q \rangle} [1 - (1 - X_p)^{q-1}], \tag{2.50}$$

and disregarding second order terms, and noticing we can start the summation at $q = 0$:

$$X_p \approx p \sum_{q=0}^{\infty} \frac{qP(q)}{\langle q \rangle} \{1 - [1 - (q-1)X_p]\} \tag{2.51}$$

$$\Leftrightarrow 1 = p \sum_{q=0}^{\infty} \frac{qP(q)}{\langle q \rangle} (q-1)X_p \tag{2.52}$$

$$\Leftrightarrow 1 = \frac{p}{\langle q \rangle} \sum_{q=0}^{\infty} q(q-1)P(q) \tag{2.53}$$

$$\Leftrightarrow 1 = \frac{p}{\langle q \rangle} (\langle q^2 \rangle - \langle q \rangle) \tag{2.54}$$

$$\Leftrightarrow 1 = \frac{p}{\langle q \rangle} (\langle q^2 \rangle - \langle q \rangle) \tag{2.55}$$

$$\Leftrightarrow 1 = p\lambda. \tag{2.56}$$

Which means that the Giant Component appears at

$$\lambda_{pcrit} \equiv \lambda = \frac{1}{p}. \tag{2.57}$$

Firstly, this result tells us that as we increase p, the smaller the transition point will get. Which makes sense as if we remove some edges when damaging the networks, it will take a higher mean degree to "compensate" for that. What happens to S_{X_p} is exactly the same as to S_X: at the critical point the mean cluster size diverges, and a solution from zero (continuous) appears. The only difference is the introduction of p, which in a way works as a "delayer" of the whole behaviour of the network. For example, what in the original graph would already constitute a Giant Connected Component, may not constitute a PGCC, as the removal of some of its edges may divide it back to smaller (finite) clusters. In our analogy, it is the inverse process of the growing fires, it is as if the forest grew back, disconnecting some of the fires. Notice also that the GCC appears when $\langle q \rangle = 1$, that is, when each node connects on average to another node. With the introduction of p, the network will reduce its number of edges, so in the Percolated Graph, this relation will not hold up. Only when we increase the mean degree so much that this relation is established again (which happens at λ_{pcrit}) do we get a Giant Component in the Percolated Graph (PGCC).

To finish, if we look at the limit for the critical point, $p = 1$, where no edges where removed, we get the original graph's value of $\lambda_{pcrit} = \lambda_{crit} = 1$, which is consistent.

A last note should be made that, from the moment we create an Erdős-Rényi graph until after we damage, each connection between two nodes will exist with probability $r \times p$. And since these probabilities are independent for each edge, and of each other, we can see the damaged network (the Percolated Graph) as an Erdős-Rényi graph of parameters (r', N), with $r' = p \times r$. Whose mean degree will be $\langle q \rangle' = \lambda' = r'N = p\lambda$.

2.2 SIR in Networks

So far we have studied the SIR Model in a fully mixed type of interaction between individuals (where everyone can meet everyone). We know than in the real world, not everyone knows everyone, as such, a proposal of a more realistic model is to use Susceptible-Infected-Recovered contagion process but in a network. Contagion processes can happen in all sorts of networks: they can happen in social networks, such as diseases or the spreading of rumor/ideas; they can happen in the Internet, such as the spreading of computer viruses [6], and so on. The type of network we will study in here are Erdős-Rényi , due to the advantages we have mentioned in the previous chapter.

The way we define an SIR model in a network is the following: firstly, we start by generating a Graph; then each node of the graph can be in either one of three states: Susceptible, Infected or Recovered, following the order of events:

$$S \to I \to R$$

We considered a process evolving in discrete timesteps, where we defined two quantities: α and β; which are the probability of recovering each timestep, and the probability of successful contagion upon making a contact (which we will define as a successful contact), respectively. At the start ($t = 0$), there is a seed, f, a fraction of nodes, chosen in a uniform random way, who are in the infected state; and the rest of the nodes are susceptible. Normally SIR Models just consider the limit $f \to 0$, whilst here, due to the model's relation with the Bootstrap Model, we will allow it to be finite, for which we will have to introduce a new definition of the Recovered Giant Connected Component. The way the system evolves is by each node making contact with all its neighbours each timestep, and if an infected node makes a successful contact with a susceptible one (probability β), the susceptible node changes to the infected state (in interactions of the type Susceptible-Susceptible, Susceptible-Recovered or Recovered-Recovered, nothing happens). Also, each infected node, each timestep, has a probability α, of jumping from an infected state to a Recovered state.

Since the evolution of the states of a node is one-directional, the dynamics in the network will stop when there are no more nodes in an infected state (no susceptible will be able to be

infected, and there are no more infected that can go to a recovered state). As such, and like in the fully mixed case, for $t \to \infty$ there will only be susceptible and/or recovered nodes.

Figure shows some typical images of the evolution of the fraction of nodes in each state over time:

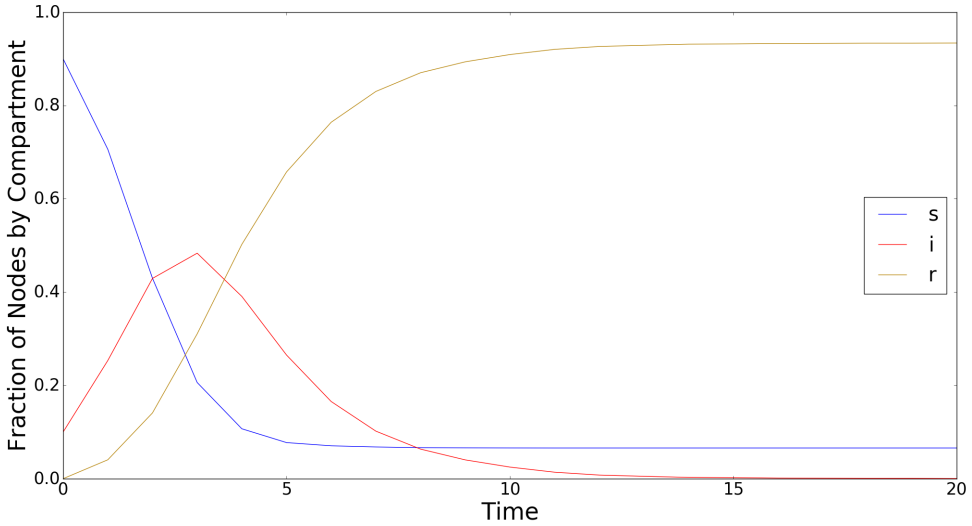


FIGURE 2.9: The evolution of the fraction of nodes in each compartment over time. This system was simulated using the algorithm described in section 4.1, for the parameters: $k = 1$, $\alpha = 0.4$, $\beta = 0.8$, $\langle q \rangle = 5$, $f = 0.1$, $N = 10000$, $N_{times} = 10$.

These graphics are very similar to the fully mixed ones, Figure 1.5. The fraction of susceptible nodes, starts as almost one (the seed was 10 % of the network), and decreases as some nodes get infected. The fraction of infected nodes, starts as the number of seeds; then there is an outbreak and the disease spreads through the network, and it increases; and finally it goes to zero, when all the nodes recover. The fraction of recovered nodes grows from zero to the total number of nodes who got infected, during the whole lifetime of the disease. Notice that since time is discrete we do not obtain smooth curves such as in Figure 1.5.

In fact, topologically, looking at the end state of the graph, if we label each node by its state (S and R), we can even see where the disease has been through since the disease leaves a "track" of recovered nodes, starting in the seeds. Actually, it is because of this that we will be able to construct quantities such as the fraction of recovered nodes in the end state.

Unlike the SIR Model for the Fully Mixed case, where we only can find relations between the equations of state, here we can obtain an exact solution for the fraction of the network that was infected, after the disease dies out, and even for the size of a Giant Connect Component constituted solely of Recovered nodes. The way this is done is by very elegantly mapping this problem to a percolation problem.

We do this by first defining the quantity p: the probability that an infected node, via a specific edge, makes at least one successful contact with the neighbour who is on the other side of

this edge, before the infected node recovers. In other words, that an edge of an infected node, is used in the transmission of the contagion agent (that we will assume is a disease, for the sake of the explanation).

Notice that all infected nodes must eventually recover for $\alpha \in]0,1]$, this means that the probability of not recovering each timestep is $(1-\alpha)$; so the probability of not recovering until time $t = t'$ is $(1-\alpha)^{t'}$ that goes to zero as $t \to \infty$.

Let's say that a node is infected at $t = 0$ (we can always set the time that the node jumped to the infected state as $t = 0$). The probability that each of its edges is used in passing the infection in the next timestep (until $t = 1$), is the probability that the node did not recover $(1-\alpha)$ (as he could not infect anyone if he had recovered) times the probability that the contact it made with its neighbour was successful β (remember that each node makes contact with all of its neighours each timestep). We define this to be $P_{t=1}$:

$$P_{t=1} = (1-\alpha) \times \beta. \tag{2.58}$$

Until $t = 2$, the probability that each edge gets used in passing the infection corresponds to the probability that it got used at $t = 1$, plus the probability that it did not get used, but the node did not recover again, $(1-\beta)(1-\alpha)$, and that it got used at $t = 2$:

$$P_{t=2} = P_{t=1} + (1-\alpha)(1-\beta)(1-\alpha)\beta = P_{t=1} \times [1 + (1-\alpha)(1-\beta)]. \tag{2.59}$$

By recurrence, one can observe that at $t = t'$, the probability that this edge ever got used is:

$$P_{t=t'} = P_{t=1} + (1-\alpha)(1-\beta)(1-\alpha)\beta + ... + [(1-\alpha)(1-\beta)]^{t'-1}(1-\alpha)\beta \tag{2.60}$$

$$= P_{t=1} \times \sum_{t''=1}^{t''=t'} [(1-\alpha)(1-\beta)]^{t''-1}. \tag{2.61}$$

So, it is equal to the sum until $t'' = t'$ that the node did not recover nor did it infect so far, but it infected in this timestep; for all timesteps.

Hence, the probability that one edge ever made a successful contact, is the limit where $t' \to \infty$:

$$p = P_{t=\infty} = P_{t=1} \sum_{t''=1}^{\infty} [(1-\alpha)(1-\beta)]^{t''-1} \tag{2.62}$$

$$= P_{t=1} \sum_{t'''=0}^{\infty} [(1-\alpha)(1-\beta)]^{t'''} \tag{2.63}$$

$$= P_{t=1} \frac{1}{1-(1-\alpha)(1-\beta)} = \frac{\beta(1-\alpha)}{\alpha+\beta(1-\alpha)}. \tag{2.64}$$

Where the change of variable $t''' = t'' - 1$ was made.

With p having been determined, it is possible to calculate the Recovered Fraction of the Network, that we will define as S_Z, in the following way:

We start by occupying the network with probability p, i.e. we assign to each edge in the network, a probability $(1-p)$ of being removed, and remove the ones selected. For a matter of clarity, like we did in the on the Percolation chapter, let's define the term Original Network (ON) to refer to the network before being damaged (the original network in which the disease actually spread); and the term Percolated Recovered Network (PRN), to the result of damaging the Original Network with probability $1-p$. We also define that a node *used* an edge, when it made at least one successful contact through it, before it recovers, independent of the state of the node in the other end. Furthermore, when we mention that an infected node "transmitted" the disease, or "passed" the infection to a neighbour we do not mean that this node was responsible for the change from the susceptible state to the infected state of the neighbour, since there could have been other infecting nodes also passing the infection to that neighbour; the same applies for the term "catched" the infection from. These terms mean that the infected node *used* an edge to this neighbour, being a sufficient condition (not necessary) to infect it. This makes each node able to be "transmitted", and "passed" the infection more than once. This also means that an infected node can transmit the disease to another infected node. For the change of state of a node from being susceptible to infected, we will say that a node was "infected".

If we look at the Original Network in the infinite time limit ($t \to \infty$), we will only observe susceptible and recovered nodes. The Percolated Recovered Network only has a significance when seen in this limit as well, since p is also an infinite time quantity. It is a representation of the end state of the system. So, and looking at this time limit, by damaging the Original Network with probability $1-p$, we obtain a network in which all the existent edges were *used*. Meaning that the nodes in the existent clusters made successful contacts with each other, before they recovered. This being, two types of clusters can appear in the PRN: a first in which at least one of the nodes in the cluster was a seed; and a second, in which none of the nodes in the cluster was a seed. Regarding the first type, since (at least) a seed existed

in the cluster, and all the edges in that cluster were *used* in passing the infection, this means that the seed passed the infection to all of its neighbours before it recovered, and that its neighbours also passed the infection to their neighbours, before they recovered. Resulting in all the nodes in clusters of this type ending up infected, and then recovered; let's call these Recovered clusters. The second case, is the case where none of the clusters' nodes was a seed. In this case, it does not matter if the edges where *used* in passing the infection or not, since there was no infection to pass in the first place. So none of the nodes in these clusters ends up infected; let's call them Susceptible Clusters.

It follows that all the recovered nodes that exist, only exist in clusters of the first type; and that the Recovered clusters are isolated from the Susceptible clusters in the PRN.

Figures 2.10 and 2.11 present the differences between the Original Network and the Percolated Recovered Network at $t = 0$ and at $t = \infty$.

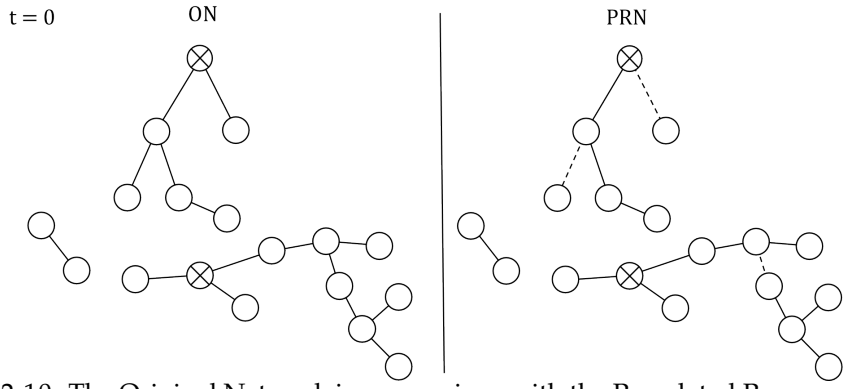


FIGURE 2.10: The Original Network in comparison with the Percolated Recovered Network at $t = 0$. The ON is shown on the left, and the PRN is shown on the right.

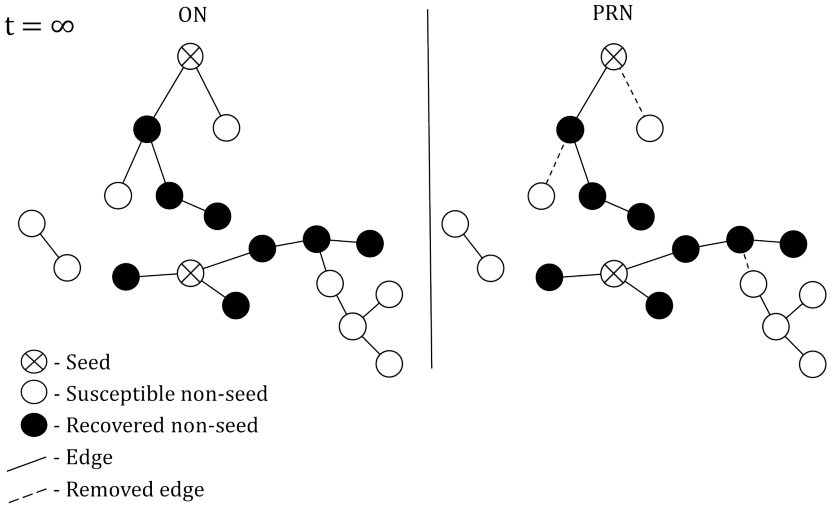


FIGURE 2.11: The Original Network in comparison with the Percolated Recovered Network at $t = \infty$. The ON is shown on the left, and the PRN is shown on the right. The definitions for the states of the nodes and edges, in the infinite time limit are presented. One can see that the clusters in which there is at least a seed, fully end up recovered since all the edges were used in passing the infection.

The way we can calculate the fraction of recovered nodes in the network, S_Z, is by calculating the sum of the sizes of these Recovered clusters (clusters made solely of recovered nodes). We can write S_Z in the following way:

$$S_z = f + (1-f)\sum_{q=1}^{\infty} P(q) \sum_{l=1}^{q} \binom{q}{l} Z^l (1-Z)^{q-l} \tag{2.65}$$

where we have introduced the quantity Z, that we define as the probability that I follow an edge and meet a node who was infected, at any time during the lifetime of the disease, and *used* this edge in transmitting the infection to me.

Regarding the expression, the first term refers to the seeds (f), since all these will eventually recover. The second term refers to the nodes that are not part of the seed, $(1-f)$, of any degree q ($\sum_{q=1}^{\infty} P(q)$), which in order to be part of the recovered component need, out of their q edges, that at least one of them leads to a neighbour which transmitted the disease to them ($\sum_{l=1}^{q} \binom{q}{l} Z^l (1-z)^{q-l}$, l is at least one). Notice that the summation over q starts at $q = 1$ as in order for a non-seed node to get infected, it needs to connect at least once to a neighbour who transmitted the disease to it. Notice also that the degree distribution used in the expression ($P(q)$) is for the Original Network, not for the Percolated Recovered Network.

Finally, we should point out that we are treating each edge that leads to a neighbour in an independent way (we write the second summation as a binomial), because we are assuming that the second moment of the distribution, $P(q)$, is finite, and that the uncorrelated random network is infinite, making it locally tree like. In disease spreading this independence of branches, allows us to "track" the disease back to the seeds, as we do in defining Z.

Notice that S_Z also represents the probability that a node, chosen uniformly at random, is recovered in the infinite time limit.

We can write Z is the following way:

$$Z = pf + p(1-f)\sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \sum_{l=1}^{q-1} \binom{q-1}{l} Z^l (1-Z)^{q-1-l}. \tag{2.66}$$

The first term represents the probability that a node is passed the infection by a seed. It is the probability that if I follow an edge (myself being the node behind that edge), that I meet a seed node (the f part of the first term), and that this edge is present in the PRN, hence that the seed transmitted me the disease before it recovered (probability p; the edge was *used*).

The second term accounts for the probability of meeting a non-seed node, but still being passed the infection through it. This is only possible if it eventually catches the disease itself,

via one of its neighbours (other than me). So, each part of the second term represents: the probability that I follow an edge and meet a non-seed node $(1-f)$; that this node has degree $q\left(\frac{qP(q)}{\langle q\rangle}\right)$, and that it has at least one neighbour (who is not me) which passed the infection to it $\left(\sum_{l=1}^{q-1}\binom{q-1}{l}Z^l(1-Z)^{q-1-l}\right)$; notice the summation over l is until $q-1$ since we are excluding the node we are coming from (myself) as a possibility); and finally that the edge was not removed when the network was damaged, i.e. that this non-seed node infected me before it recovered (the factor p). To cover all the possibilities of the degrees of non-seed nodes, we have to sum over all values of q, starting in $q=2$ since a non-seed node needs to have at least two edges, one to connect (and transmit the disease) to me, and another to catch the disease itself. Once again, $P(q)$ is the degree distribution in the Original Network.

For a better understanding, figure 2.12 shows all the possible cases of Z, and $1-Z$.

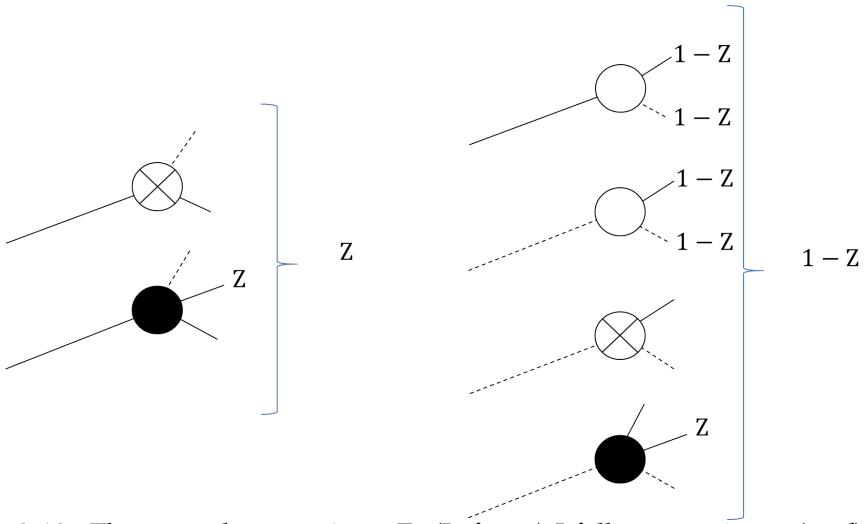


FIGURE 2.12: The cases that constitute Z: (Left top) I follow an existent (*used*) edge, and meet a seed; (Left bottom) or that I follow an existent (*used*) edge and meet a non-seed node which eventually was passed the infection via somewhere else. The cases that constitute $1-Z$ are the cases in which I follow an edge and meet: (right top) a non-seed node who never got infected during the lifetime of the disease, and this edge was not removed (so even though the edge was *used* in transmitting the disease, there was no disease to transmit in the first place); (right second) a non-seed node who never got infected, and the edge was removed (meaning that even if he did, he would not infect me); (right third) a seed, but this edge was removed (so it did not infect me); and (right bottom) a non-seed node which (eventually) got passed the infection, but this edge was removed (so it means did not infect me). In these diagrams, other than the previous definition used for the state of each node in the infinite time limit, we also define that a node which has all its edges connecting to a presented quantity (Z or $1-Z$ in this case) must necessarily have its edges leading to that quantity; if a node has edges that do not have a defined quantity in the other end, these edges can connect to anything. This definition only applies to the edges on the right side of each node, as the edges on the left are representing where we are coming from (the node in which we are applying this quantity).

Notice that building Z in this way is only possible since each branch is independent, and the probability that I get infected via one neighbour is independent of weather or not my others neighbours where passed the infection. This is also the reason we can say we "track

the disease", because if I got infected via one brach, either my immediate neighbour is a seed and infected me, or it is not a seed but it (eventually) got the infection via one of its (other-than-me) neighbours and then infected me. And for it to get the infection, this argument can be applied as well, either one of its immediate (other-than-me) neighbours was a seed and infected it, or none of its neighbours was a seed but at least one of them got the infection via one of their own (other-than-it) neighbours and infected it, and so forth. Until, eventually one of those non-seed nodes meets a seed. We are going from node to node, until we meet the seeds from which the disease spread, "tracking" the disease.

Also, since Z is built in this way, in the limit $t \to \infty$, it represents the probability that upon following an edge, in the Original Network, that I meet what will be a cluster of recovered nodes, in the Percolated Recovered Network. Because it is implicit in Z that there is a seed in that branch (That will later become a part of a recovered nodes' cluster). The other clusters, even though they connect between themselves, they do not have a seed to originally spread the disease, so all nodes will remain susceptible.

Before we move on, let us just notice that in building Z we assumed a uniform distribution for the initial fraction of infected nodes, f, the seeds (i.e. each node had the same probability of starting as a seed). As such, and since we are in an infinite network, if we group the nodes by their degree, all the groups will have the same fraction f of seeds. If we were to choose a different distribution, say f_q, where each group of nodes with the same degree, would have a different number of seeds, we would have to write

$$Z = p\sum_{q=1}^{\infty}\frac{qP(q)}{<q>}f_q + p\sum_{q=2}^{\infty}\frac{qP(q)}{<q>}\sum_{l=1}^{q-1}(1-f_q)\binom{q-1}{l}Z^l(1-z)^{q-1-l}, \tag{2.67}$$

with f_q being the fraction of seeds in the group of nodes of the same degree q.

Equation (2.66), can be simplified in the following way:

$$Z = pf + p(1-f)\sum_{q=2}^{\infty}\frac{qP(q)}{<q>}\sum_{l=1}^{q-1}\binom{q-1}{l}Z^l(1-Z)^{q-1-l} \tag{2.68}$$

$$Z = pf + p(1-f)\sum_{q=2}^{\infty}\frac{qP(q)}{<q>}[1-(1-Z)^{q-1}], \tag{2.69}$$

where we used the binomial theorem (Equation (2.15)) in the second summation. Then

$$\sum_{l=1}^{q-1}\binom{q-1}{l}Z^l(1-z)^{q-1-l}=\sum_{l=0}^{q-1}\binom{q-1}{l}Z^l(1-Z)^{q-1-l}-(1-Z)^{q-1}=1-(1-Z)^{q-1}. \tag{2.70}$$

For an Erdős-Rényi graph, we can write:

$$Z=pf+p(1-f)\sum_{q=1}^{\infty}\frac{qe^{-\lambda}\lambda^q}{<q>q!}[1-(1-Z)^{q-1}]. \tag{2.71}$$

Notice that we can start the summation in $q=1$, as that term is zero. Since $<q>=\lambda$,

$$Z=pf+p(1-f)\sum_{q=1}^{\infty}\frac{e^{-\lambda}\lambda^{(q-1)}}{(q-1)!}[1-(1-Z)^{q-1}] \tag{2.72}$$

$$=pf+p(1-f)\sum_{q=1}^{\infty}\frac{e^{-\lambda}\lambda^{(q-1)}}{(q-1)!}-p(1-f)\sum_{q=1}^{\infty}\frac{e^{-\lambda}\lambda^{(q-1)}}{(q-1)!}(1-Z)^{q-1}. \tag{2.73}$$

Performing a change of variable $q'=q-1$, we get:

$$Z=pf+p(1-f)\sum_{q'=0}^{\infty}\frac{e^{-\lambda}\lambda^{q'}}{q'!}-p(1-f)\sum_{q'=0}^{\infty}\frac{e^{-\lambda}\lambda^{q'}}{q'!}(1-Z)^{q'}. \tag{2.74}$$

The first summation is just one, as it is $\sum_{q'=0}^{\infty}P(q)=1$, since a probability distribution is normalized to one. The second summation, we can treat it in order to make it a Poisson distribution of variable $\lambda'=\lambda^{q'}(1-Z)^{q'}$. Let us make this substitutions:

$$Z=pf+p(1-f)-p(1-f)\sum_{q'=0}^{\infty}e^{-\lambda Z}\frac{e^{-\lambda(1-Z)}\lambda'^{q'}}{q'!} \tag{2.75}$$

$$Z=pf+p(1-f)-p(1-f)e^{-\lambda Z}\sum_{q'=0}^{\infty}\frac{e^{-\lambda'}\lambda'^{q'}}{q'!}. \tag{2.76}$$

If we notice, we get again a summation over all the possible values of q', in a Poisson distribution of variable λ', which is also one. Hence we obtain:

$$Z = pf + p(1-f) - p(1-f)e^{-\lambda Z} \tag{2.77}$$

$$\Leftrightarrow Z = p\left[1 - (1-f)e^{-\lambda Z}\right]. \tag{2.78}$$

This transcendental equation can be solved using the same method as we did on the Percolation section 2.1.2: by defining each side of the equation as independent functions.

In this case, the Left Hand Side equation, LHS, and the Right Hand Side, RHS, will look as shown in figure 2.13.

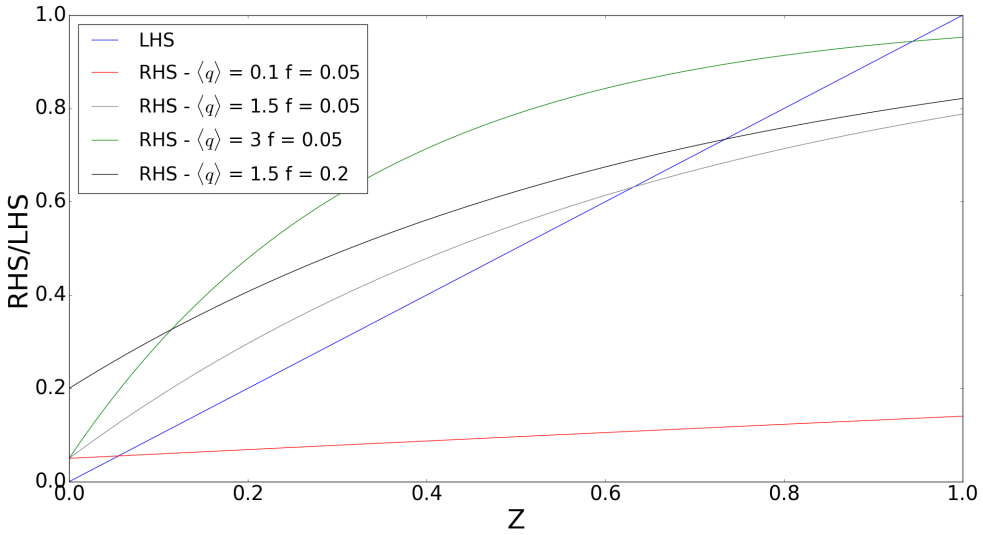


FIGURE 2.13: The shape of the RHS function for different values of $\langle q \rangle$. The limits of the figure are $Z \in [0, 1]$ since it is the regime where the physical solutions exist. The solutions for Z are when RHS intersects with the LHS function. The value used for p and f were: $p = 0.7$, $f = 0.05$ and $f = 0.2$, and the values of the mean degree were: $\langle q \rangle = 0.1$, before the percolation threshold; $\langle q \rangle = 1.5$, right after the percolation threshold; $\langle q \rangle = 3$, a lot after the percolation threshold. We can see that as the mean degree increases, so does the solution for Z, meaning that each node has a higher probability of getting infected. The minimum of the RHS functions is f, the point in which they start at. This can be seen in comparing both the lines of $\langle q \rangle = 1.5$.

In the case of Z, for all values of $< q >$, we only have a global stable solution, so it does not matter where we set the initial conditions (Z_0), because we will always reach it.

Having calculated Z, we can obtain the value of S_z, by just substituting Z into Equation (2.65). Before doing that, however, we can simplify it:

$$S_z = f + (1-f)\sum_{q=1}^{\infty} P(q) \sum_{l=1}^{q} \binom{q}{l} Z^l (1-z)^{q-l} \tag{2.79}$$

$$= f + (1-f)\sum_{q=1}^{\infty} P(q)[1-(1-Z)^q]. \tag{2.80}$$

Notice we can start the summation from $q = 0$:

$$S_z = f + (1-f)\sum_{q=0}^{\infty} P(q) - (1-f)\sum_{q=0}^{\infty} P(q)(1-Z)^q \tag{2.81}$$

$$= f + (1-f) - (1-f)\sum_{q=0}^{\infty} \frac{e^{-\lambda}\lambda^q}{q!}(1-Z)^q \tag{2.82}$$

$$= f + (1-f) - (1-f)e^{-\lambda Z}\sum_{q=0}^{\infty} \frac{e^{-\lambda(1-Z)}(\lambda(1-Z))^q}{q!} \tag{2.83}$$

$$\Leftrightarrow S_z = 1 - (1-f)e^{-\lambda Z}. \tag{2.84}$$

Figure 2.14 shows the evolution of S_z with the mean degree, for different values of p and f.

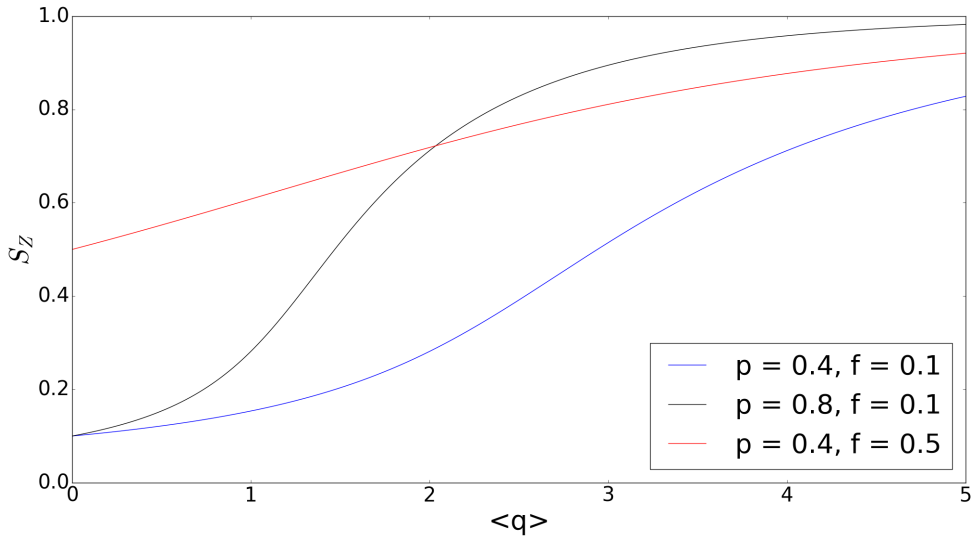


FIGURE 2.14: The figure plots the evolution of S_z with the mean degree, for different values of p and f.

First of all, we can see that as $< q >$ increases, so does the final recovered component; which is rather intuitive, as if every node has more neighbours, they are more likely to be infected. In other words, they are in a "shorter distance" from the seed, and the probability that each node connects to a seed, or to someone who is close to one, increases.

Secondly, we can observe that for smaller values of p, the disease reaches a smaller fraction of the network. This is also intuitive, as smaller values of p are associated with higher recovery rates (α) and/or smaller infection rates (β), making it more difficult for each node to infect someone before it recovers. In our analogy to the Percolated Recovered Network, a smaller number of edges will exist, reducing the number of recovered nodes. Since the number of seeds is the same, each of them will infect an entire smaller cluster, resulting in smaller recovered clusters.

Lastly, S_Z grows also with the value of the seed. This makes sense since not only will there be a starting contribution of more nodes, but also since these exist in greater number the mean distance between a susceptible node and a seed is reduced.

2.2.1 The Classic Recovered GCC

An interesting quantity that can appear in a SIR model is a Giant Connected Component made solely out of recovered nodes, a RGCC. Just like the GCC in the Random Networks section, the RGCC will take a finite portion of the infinite network. Notice that the RGCC is part of the GCC (RGCC $\in$ GCC), that is, the nodes that constitute a RGCC will also constitute a GCC, since a GCC has to exist in the first place, so that the disease can spread and infect a part of it.

Disease spreading (and contagion models) is a good example of how the GCC is where "most of the action will take place", as it is where the disease is offered the possibility to infect more nodes in a row. Also, since the component is all connected, from the moment that a seed exists, everyone is potentially at risk (all the infinite nodes in the GCC). Furthermore, the study of this quantity, and consequently of the RGCC, is important nowadays, since the human network has some characteristics of a Giant Connected Component: it is completely connected (there are no isolated people anymore), and it is made by a very large number of people ($> 7 \times 10^9$). Also, regarding disease prevention, the RGCC it is also where we can measure the worse impact of the disease, and test the best strategies against its propagation.

Classically, the RGCC is defined as the biggest Recovered cluster in the Percolated Recovered Network. So, for a matter of clarity, when we refer to the CRGCC from here henceforth, we are referring to the classical definition of the RGCC. Which we present in this chapter.

Figure 2.15 shows how the RGCC looks in the Original Network, compared to the Percolated Recovered Network:

The way the size of the Classic Recovered GCC is calculated, is by building an argument similar to the one we did for S_Z:

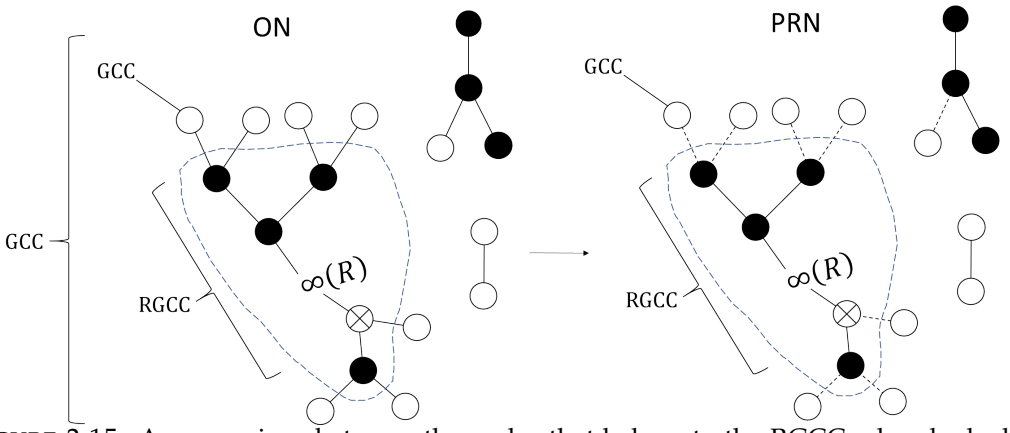


FIGURE 2.15: A comparison between the nodes that belong to the RGCC when looked at in the ON, and in the PRN. The Original Network is shown on the left, and the Percolated Recovered Network on the right. In the original network it is clear that the nodes that belong to the RGCC also belong to the GCC.

$$S_{X_C} = f \sum_{q=1}^{\infty} P(q) \sum_{l=1}^{q} \binom{q}{l} X_C^l (1 - X_C)^{q-l} + (1-f) \sum_{q=1}^{\infty} P(q) \sum_{l=1}^{q} \binom{q}{l} (1-Z)^{q-l} \sum_{m=1}^{l} \binom{l}{m} X_C^l (Z - X_C)^{l-m}. \tag{2.85}$$

Where we define X as the probability, if I am a node, that I follow an edge and meet a node who passed the infection to me, and that it belongs to the Recovered GCC. Notice that $X \subset Z$, since all the nodes who passed the infection to me and connect to the CRGCC (X), necessarily belong to the group of nodes who passed the infection me (Z).

We should make the following disclaimer: X_C represents the probability of a certain configuration of the network occurring. However, for convenience in some discussions, we will sometimes use "X_C" to also refer to the set of cases when this configuration occurs (similarly for Z and to future probabilities that may come up, referring to probabilities of sets of cases of following edges). In this way we may write, for example, $X_C \subset Z$, as the configuration that is satisfied with probability X_C, that always also satisfies the configuration Z.

The first term in Equation (2.86) refers to the seeds, this is why there is a factor of f multiplying. In order for a seed to belong to the CRGCC, since they will end up being recovered already, they just need to connect to someone who belongs to the Classic Recovered Giant Connected Component (they just need the a connection to exist in the Percolated Recovered Network). This is represented by the second summation; the factor $P(q)$ accounts for the probability that the seed has q neighbours, and the first summation accounts for all the possibilities of q. We start the summation over q in $q = 1$, as a seed needs to have one edge to

connect (and belong) to the Classic Recovered GCC. The second term accounts for the possibility of a node not being a seed, $(1-f)$; that it has any degree q, $\sum_{l=1}^{q} P(q)$; that at least one of its neighbours passed the infection to it (Z), and of those that at least one connects to the CRGCC (X).(All of this is represented by $\sum_{l=1}^{q} \binom{q}{l}(1-Z)^{q-l} \sum_{m=1}^{l} \binom{l}{m} X_C^l (Z-X_C)^{l-m}$: of its q neighbours, l leaded to someone who transmitted the disease it, and of those, at least $m=1$ of them connect to the CRGCC).

As for X_C, it can be defined as for Z, but adding the condition that at least one of those nodes who passed the infection to me, is part of the RGCC:

$$X_C = pf \sum_{q=2}^{\infty} \frac{qP(q)}{<q>} \sum_{l=1}^{q-1} \binom{q-1}{l} X_C^l (1-X_C)^{q-l-1} \tag{2.86}$$
$$+ p(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{<q>} \sum_{l=1}^{q-1} \binom{q-1}{l} (1-Z)^{q-1-l} \sum_{m=1}^{l} \binom{l}{m} X_C^l (Z-X_C)^{l-m}.$$

The first term refers to the contribution of following an edge and meeting a seed who belongs to the CRGCC. It reads as the probability that, if I am a node behind the edge which we are following, that this edge was not removed during the damaging of the network, p (so, that the seed *used* this edge in passing the infection me), that the node I meet is indeed a seed (f), of any degree q ($\sum_{q=2}^{\infty} \frac{qP(q)}{<q>}$), who has at least one edge (which is not the one I am coming from) that connects to the Classic Recovered GCC ($\sum_{l=1}^{q-1}$). Once again, the summation ends at $(q-1)$ since in order for X to be a sufficient condition for me to connect to the CRGCC, the node I meet, as to connect to the CRGCC itself via of of its neighbours who is not me).

The second term refers to the possibility that I meet a node who is not a seed, and still connect of the CRGCC. For this to happen the node I meet has to catch the disease at some point, and connect to the CRGCC. In order to fulfill these conditions we need: reaching a non-seed node $(1-f)$; of any degree q ($\sum_{q=2}^{\infty} \frac{qP(q)}{<q>}$); who has at least one neighbour who passed the infection to it, and of those neighbours, at least one of them connects to X_C (represented by the summations over l and m respectively); and of course, that the edge I am following exists in the PRN p (i.e. that node passed the infection to me before it recovered).

Notice that both summations start in $q=2$ as the node I meet needs one edge to pass the infection to me, and another to connect to the CRGCC.

Figure 2.16 shows the possible cases of X_C, in which is possible to see that X_C is part of Z. Implying that being in X_C is a sufficient condition to be infected. This figure also shows the cases of $1-X_C$.

We can treat the equations for X_C (and of S_{X_C}) in a similar way that we did for Z:

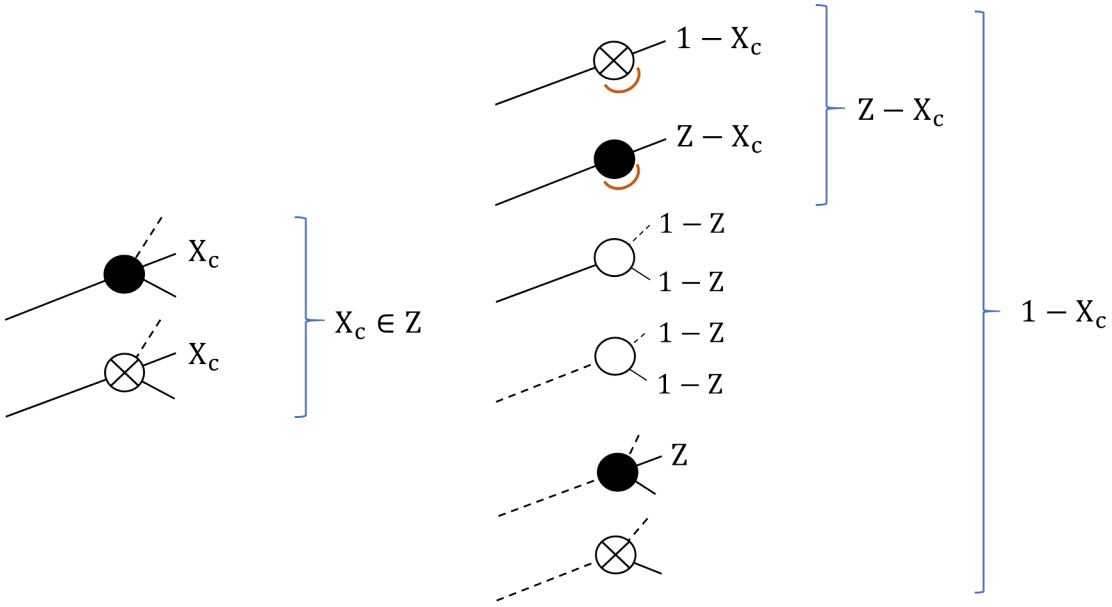


FIGURE 2.16: The cases that constitute X_C: (Left) I follow an existent (*used*) edge and meet a node (seed or non-seed) that passed me the infection, which connects to the CRGCC. The cases that constitute $1 - X_C$: (Right first and second) I follow an existent (*used*) edge and meet a node (seed or non-seed) that passed me the infection, which does not connect to the CRGCC; (Right third and fourth) I follow an edge and meet a non-seed node that did not pass me the infection, (whether the edge was *used* or not); (Right fifth and sixth) I follow a non-existent (not *used*) edge and meet a node (seed or non-seed) that did not passed me the infection, whether it connects to the CRGCC or not. This figure uses the same definitions as Figure 2.12 .

$$X_C = pf \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} [1 - (1 - X_C)^{q-1}] \tag{2.87}$$
$$+ p(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \sum_{l=1}^{q-1} \binom{q-1}{l} (1-Z)^{q-1-l} [Z^l - (Z - X_C)^l],$$

where we used the binomial theorem.

$$X_C = pf \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \left[1 - (1 - X_C)^{q-1}\right] + p(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \left[\sum_{l=1}^{q-1} \binom{q-1}{l} (1-Z)^{q-1-l} Z^l \right. \tag{2.88}$$

$$\left. - \sum_{l=1}^{q-1} \binom{q-1}{l} (1-Z)^{q-1-l} (Z - X_C)^l \right]$$

$$= pf \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} [1 - (1 - X_C)^{q-1}] \tag{2.89}$$

$$+ p(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \left\{1 - (1-Z)^{q-1} - [(1-X_C)^{q-1} - (1-Z)^{q-1}]\right\}.$$

Again, we can start the summations in $q = 1$:

$$X_C = pf \sum_{q=1}^{\infty} \frac{qP(q)}{< q >} [1 - (1 - X_C)^{q-1}] + p(1-f) \sum_{q=1}^{\infty} \frac{qP(q)}{< q >} [1 - (1 - X_C)^{q-1}]. \tag{2.90}$$

Perceiving that the summations over q are the same:

$$\Rightarrow X_C = p \sum_{q=1}^{\infty} \frac{qP(q)}{< q >} \left[1 - (1 - X_C)^{q-1}\right]. \tag{2.91}$$

If we notice, this equation has the same structure as the second term in Equation (2.69) (because the term $q = 1$ is zero), with the substitution X_C instead of Z, and p for $p(1-f)$. Meaning we can adapt the result using this substitution, obtaining

$$X_C = p(1 - e^{-\lambda X_C}). \tag{2.92}$$

Which has a very familiar form as the equations we have derived so far. In fact, it is the exact same equation as the Equation (2.47) for the size of the PGCC, in the Percolation subsection, 2.1.2, by performing the substitution $X_p \to X_C$. So we can solve it in the exact same way.

Since the solutions are the same we know (in advance) that the trivial one $X_C = 0$ is stable before a critical point, $\langle q \rangle < \lambda_{critC}$, and becomes unstable after it, coinciding with the appearance of a second stable solution responsible for the appearance of the CRGCC, of critical point

$$\lambda_{critC} = \frac{1}{p}. \tag{2.93}$$

This behaviour is shown in Figure 2.8.

Having the solution for X, one can now calculate the actual fraction of the network that belongs to the Recovered GCC:

$$\begin{aligned} S_{X_C} &= f \sum_{q=1}^{\infty} P(q) \sum_{l=1}^{q} \binom{q}{l} X_C^l (1-X_C)^{q-l} \qquad (2.94)\\ &+ (1-f) \sum_{q=1}^{\infty} P(q) \sum_{l=1}^{q} \binom{q}{l} (1-Z)^{q-l} \sum_{m=1}^{l} \binom{l}{m} X^l (Z-X_C)^{l-m} \\ &= f \sum_{q=1}^{\infty} P(q)[1-(1-X_C)^q] + (1-f) \sum_{q=1}^{\infty} P(q) \sum_{l=1}^{q} \binom{q}{l} (1-Z)^{q-l}[Z^l - (Z-X_C)^l] \qquad (2.95)\\ &= f \sum_{q=1}^{\infty} P(q)[1-(1-X_C)^q] + (1-f) \sum_{q=1}^{\infty} P(q)[1-(1-X_C)^q]. \qquad (2.96) \end{aligned}$$

Notice, again, that all the summations can start in $q = 0$:

$$S_{X_C} = f \sum_{q=0}^{\infty} \frac{e^{-\lambda}\lambda^q}{q!}[1-(1-X_C)^q] + (1-f) \sum_{q=0}^{\infty} \frac{e^{-\lambda}\lambda^q}{q!}[1-(1-X_C)^q], \tag{2.97}$$

and by using the same trick as before, we get:

$$\begin{aligned} S_{X_C} &= f(1-e^{-\lambda X_C}) + (1-f)(1-e^{-\lambda X_C}) \qquad (2.98)\\ S_{X_C} &= 1 - e^{-\lambda X_C}. \qquad (2.99) \end{aligned}$$

Figure 2.17 shows the evolution of S_{X_C} with $\langle q \rangle$.

We can observe that when $\langle q \rangle > \frac{1}{p}$ a Recovered Giant Connected Component appears. This behaviour corresponds to the appearance of the second solution in X_C, and to a second order phase transition.

We can also see that as p decreases, λ_{critC} increases, deviating more and more from the percolation threshold (which for a non-damaged Erdős-Rényi graph is $\angle q \rangle = 1$). In terms of disease spreading, this deviation is due to the existence of an α and a β. If we are in the

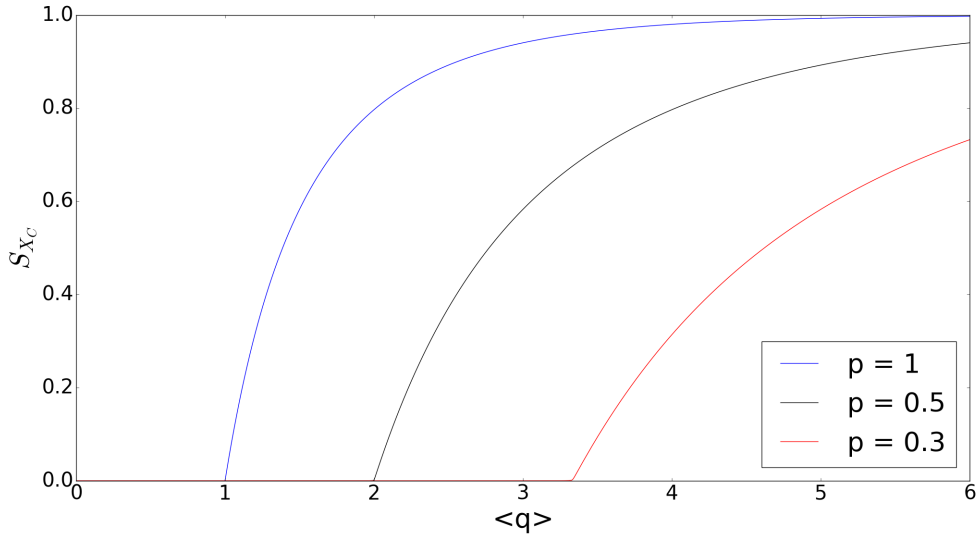


FIGURE 2.17: The evolution of S_{X_C} with $\langle q \rangle$, for different values of p. The blue line represents the classic percolation threshold for a non-damaged Erdős-Rényi graph, $p = 1$.

regime $\langle q \rangle < \lambda_{critC}$, even though there might be a GCC, the seeds that exist will only be able to contaminate a few of their neighbours, not an infinite fraction. But, as $\langle q \rangle$ increases, they will be able to infect more and more neighbours, and these neighbours of them as well, until the whole Giant Component is infected, coinciding with the GCC.

In terms of network structure, this phase transition happens in the same way as the Percolation phase transition, only this time made solely by Recovered Clusters. So, if we look in the infinite time limit, as we increase $\langle q \rangle$, the Recovered clusters will grow and grow, but being finite when $\langle q \rangle < \frac{1}{p}$. After this, they start connecting to each other, and at $\langle q \rangle = \frac{1}{p}$, their expected size diverges, making an infinite portion of the network (like the analogy we made of the fires in the forest). As p decreases, more edges will be removed, and so this behaviour only happens for bigger $\langle q \rangle$.

Another thing we can notice is that the Giant Connected fraction of Recovered nodes does not depend on the fraction of seeds. This means that after $\langle q \rangle > \frac{1}{p}$, an outbreak in the population will happen, and also, that the size of this outbreak will be the same, whether we start with a big portion of the population infected, or with just a very small fraction of it. Furthermore, the only parameter that decides how dense the network needs to be to exist an outbreak is p, a relationship between α and β.

2.3 The G-GCC

The G-GCC stands for Guilherme/Gareth - Giant Connected Component. The G-GCC represents a new way of counting the nodes that generate a Recovered Giant Connected Component, and in our opinion, a more intuitive one. If we think about what the Classical Recovered GCC means, we get to the conclusion that it represents the biggest connected component in the Percolated Recovered Network, i.e. a component in which all edges were *used* in transmitting the disease, and in which there was at least one seed; so a component that has all its nodes infected. At first glance, this might seem an accurate definition, but we argue that, firstly not always the biggest recovered component in the PRN is the same as the one in the ON (is not the RGCC), nor does it keeps track of the disease spread correctly; and secondly, that the CRGCC has some conceptual ambiguity, excluding some nodes from it, but including others of the same sort.

This ambiguity and discrimination exists when two nodes end up recovered and are neighbours in the Original Network, but did not get infected via each other. Notice that the existence of an edge in the PRN (the *usage* of an edge), between nodes who did not get infected by each other, but ended up recovered, is redundant to the disease spreading. It has no implication on the nodes who get infected, nor of how the disease spread through the network. Despite this edge having no importance, the existence of this edge in the PRN can determine whether a node belongs to the CRGCC or not, based on a probability p. Furthermore, the existence of this edge is rather ambiguous, since it implies that an infected node transmitted the disease to an already infected, or even to a recovered node.

We present a couple of cases, that we believe make our case clearer.

The first case is the following: Let us assume that there are three nodes, A, B and C in the network, who are seeds. Let us also say, that A is a neighbour of B, and of C, in the Original Network. Let us assume also that node A connects to the GCC. Let us define as a1, b1 and c1, the branches of infected nodes that started in the seeds A, B and C respectively, after the disease died. Let us say also, that in the infinite time limit there is a CRGCC and that seed A connects to it. In this case, since nodes B and C, are its neighbours, and seeds, they should also be part (as well as their recovered branches, b1 and c1) of the RGCC. However, in the CRGCC these seeds only have a probability p of being included in it. Figure 2.18 shows this case presented in the Original Network, and the Percolated Network, in the infinite time limit.

In fact, for every N edges of this sort, pN of them would be kept, and $(1-p)N$ would be removed. Let us say, that upon damaging the network, the edge A-B was kept (representing the pN connections that would be kept), and edge A-C was not (representing the others). In this case, the whole branch c1, which, in the ON would be part the RGCC, would be excluded

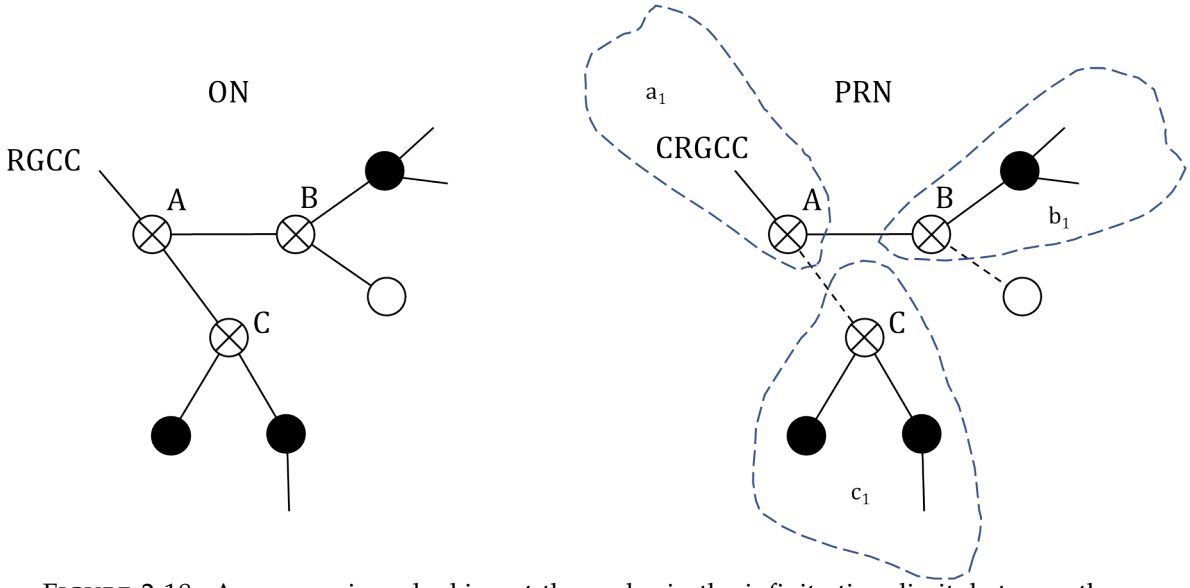


FIGURE 2.18: An comparison, looking at the nodes in the infinite time limit, between the Original Network, and the Percolated Recovered Network. Nodes A, B and C are seeds and neighbours in the Original network, yet not all of them connect to the CRGCC.

in the CRGCC, whilst branch b1 would be included in both. Also, the inclusion/exclusion of these edges, offers some conceptual ambiguity, because what does it mean to keep an edge of this sort? We know that p, in the CRGCC, represents the probability that each edge gets *used* in transmitting the disease, but what does it mean to say that an infected node (in this case a seed) transmitted the disease to an already infected node (the other seed)? And, more importantly, does it make sense to distinguish identical connections such as those of the A-C type, from those of A-B type, including one and excluding the others?

We do not think so, especially having noticed that the existence or not of these edges does not make a difference in the spreading of the disease, and as a consequence, neither does it in the number of nodes who get infected. So, there is no motive for this segregation. And in terms of what it means for an infected node to infect an already infected node, we do not believe this has any physical relevance, so it should be indifferent for the size of the RGCC whether this edge exists or not.

This might seem as a very specific example, but it is in fact a specific case of a much broader family of counter-examples to the CRGCC. This family is constituted by all nodes who ended up recovered, and have recovered neighbours in the ON, but that did not get infected via each other.

The second example we present is the following: let us say that there are three finite clusters C1, C2 and C3 that have at least one seed, which in the infinite time limit are made solely of recovered nodes. Let us also say that C1 connects to C2 and to the CRGCC, as shown in Figure 2.19.

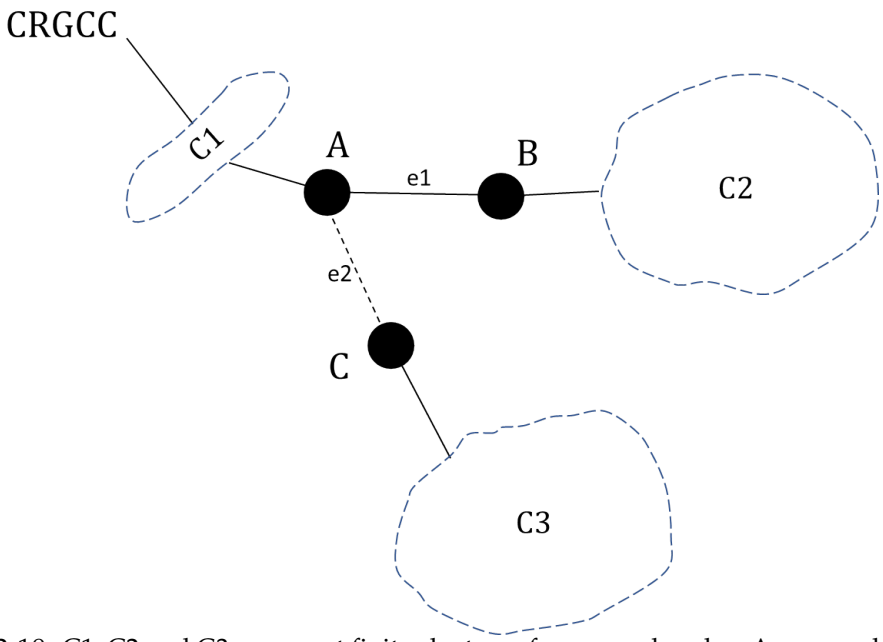


FIGURE 2.19: C1, C2 and C3 represent finite clusters of recovered nodes. An example where the CRGCC makes the exclusion of node C and C3, but includes node B and C2

Notice that in the path that connects C1 to C2 in the PRN (in fact, in the path for any connection of two seeds), there had to be, at least two nodes that *used* an edge, even though they did not infect through it (they did not need it to end up recovered). Connections of this sort can be either to seeds, or to non-seed nodes that got infected via another path. Notice that for N of these cases, pN of them will have an edge in the PRN, represented by node A, node B and the edge e1; and $(1-p)N$ will not, represented by node A, node C and edge e2. Notice that node A, B and C can either be non-seed nodes or seeds, in fact, if all of them are seeds, we obtain the previous example.

Firstly, notice that in the ON all these nodes belong to the same component, to the RGCC, however, only node A and node B belong to the CRGCC. Note also that node A connects to both node B and C1, meaning, from our definition of an edge existing in the PRN, that both node B and (a node of) C1 made a successful contact in transmitting the disease to node A. Once again, the physical significance of this is redundant. This is because during the spreading of the disease, either (the node from) C1 infected node A first, or node B did. As we assumed before, let's say C1 infected node A first and only then did node B transmitted the disease to node A (we would arrive to the same conclusion if we had assumed the opposite). What does the existence of e1 mean in this case? Because it either means that node B passed the infection to an already infected node, or worse, in the case that node A had already recovered, it means that node B transmitted the disease to an already recovered node! Another interpretation is that it was node A who *used* this edge to transmit the disease to the already infected node, node B.

This is another example where e1 is a redundant edge, since it did not infect, even though it exists in the PRN. It also was not important for the spreading of the disease, since all the nodes that ended up recovered, would end up recovered regardless of its existence. However, despite the not importance of this edge, node B and C2 are included in the CRGCC, whilst, node C and C3, who also did not infect node A, are not.

We note, that this configuration does not represent C1 and node B infecting node A at the same time. We are not saying that it cannot happen, we are saying that damaging the network with probability p does not represent this. Notice that p is a quantity of a stable state system, when $t \longrightarrow \infty$. So the existence of e1 and e2, means that both edges got *used* in transmitted the disease before each infected node recovered, not that they got *used* at the same time. In order to construct the probability that node A got infected at the same time by C1 and node B one would have to construct a whole argument over the time dynamics of the network, since the node from C1 that infected node A did probably not get infected at the same time as node B. In the specific case that they did, or that they are both seeds, it is actually possible to do so, since we can define the time that they started to be infected as $t = 0$. By using a same argument as we used to construct p, we can arrive at the conclusion that the probability that both edges got *used* at the same time is $p_{sametime} = \frac{[\beta(1-\alpha)]^2}{1-[(1-\alpha)(1-\beta)]^2} \leq \frac{[\beta(1-\alpha)]^2}{[1-(1-\alpha)(1-\beta)]^2} = p^2$, which is the probability that both e1 and e2 exist, and proving that they are not the same quantity.

To sum up, these cases do not affect the nodes who end up recovered, however there is an inclusion of some and exclusion of others of the CRGCC, based on a probability p, that has some ambiguity associated with it, and not in the actual path of the infection. Making the CRGCC not an accurate representation of the RGCC, nor of the path of the disease.

To finish, we should note that these cases cannot be disregarded in order to predict correctly the size of the RGCC (i.e. the number cases like this does not go to zeros as $N \longrightarrow \infty$), as is shown by our theory, together with numerical results of the G-GCC.

Our main concern when addressing this problem was coherence. So, when facing cases of this sort, we think that one should either keep all edges that connect recovered nodes in the ON, or reject all of them; depending on what we are trying to measure. If we are trying to measure the seed that was responsible for the infection of a greater number of nodes, we should opt for the latest. However, if we are looking at the Original Network and measuring the biggest number of nodes, who are neighbours, that ended up recovered, i.e. the RGCC, we propose keeping all these edges. And this is what the G-GCC represents, the biggest number of nodes that in the Original Network (reality if I dare), ended up recovered and are connected, that also make an infinite fraction of it.

We opted for creating a G-GCC because we believe this is of more usefulness than creating a, let's define it as Used-GCC, in which only edges which were actually *used* in infecting nodes are kept. We believe this for the following reasons:

-Even though in building a Used-GCC we could actually get the biggest cluster that got infected by a single seed, it could be misleading. This is because, for higher values of f, the Used-GCC would be smaller since there are more seeds competing for the susceptible nodes. Thus, each seed would only infect some nodes around it, because after this, the nodes that got infected originally by it, would quickly meet other nodes who had already been infected by other seeds, and could not infect anyone anymore. And the fact that the Used-GCC is smaller could be misleading us into thinking that hardly anyone got infected in the GCC, when the opposite happened. In fact, in the limit $f \longrightarrow 1$ we would get the result that the size of the Used-GCC would be zero, however everyone ended up recovered. Thus, the Used-GCC could sometimes not detect the existence of an outbreak, making it not sufficient to study the whole behaviour of the network, nor of the RGCC.

-Also, a small ambiguity also would happen when deciding which component does a node, which got infected by two different nodes at the same time, belong to. Although, this "ambiguity" could be quickly solved, by, for example, attributing the double infected node to both the components in separate (i.e. attribute it to each component, but not connect the both components; otherwise we would get a pseudo-CRGCC that would include nodes that got infected by different paths and at the same time!), it would require some consideration.

Before explaining how the G-GCC is built, we should point out that it is already possible to observe some conceptual ambiguity when defining X_C. Notice that Z represents the probability that, being myself a node, that I follow an edge, and meet a neighbour who passed the infection to me. When calculating S_Z, this probability is only applied to non-seed nodes, correctly, since they are the only ones who can get infected. However, when building X_C, we define it as not only the probability that I follow an edge, and meet a neighbour who passed the infection to me, but also that this node connects to the CRGCC. Notice that X_C, is a subset of Z. When we calculate the S_{X_C}, Equation (2.86), we use X_C, that just like Z we should only apply to non-seed nodes, to represent the probability that a seed connects to the CRGCC via one of its edges. One could notice immediately that this is conceptually ambiguous, since we are requiring for a seed to belong to the CRGCC, that it has to be passed the infection via at least one of its edges. (Let us just state that X_C is not the probability that the seed infected its neighbours, since Z, of which X_C is a subset, represents that the probability of a node to *GET* passed the infection via one of its edges). The cases that this way of defining X_C misses are related to X_b, that we present in section 3.2.1.

In what concerns Z, the only cases that it includes that are part of this ambiguity, is when non-seed nodes get passed the infection via two different paths. However, this does not

refute the validity of S_Z, since the existence or not of these redundant edges, is indifferent to how many of them are recovered. S_Z will still represent the fraction of recovered nodes. This is also the reason why we can still use $(1 - Z)$ in our calculations: it still represents the probability of a non-seed node of either having all its edges removed (not possibly being able to have been infected), or that if its edges did not get all removed, the ones that didn't connect to a Susceptible cluster in the PRN.

2.3.1 Calculation of the G-GCC

The way we construct the G-GCC is also working on the PRN (the network with occupied edges with probability p), checking if each edge was removed or not, but including all the cases of recovered nodes which connect to recovered nodes, that the Classical RGCC is missing.

Let us define X_G as the probability that, following an edge from a recovered node, we connect to the G-GCC. Figure 2.20 presents the possible cases in which this can happen.

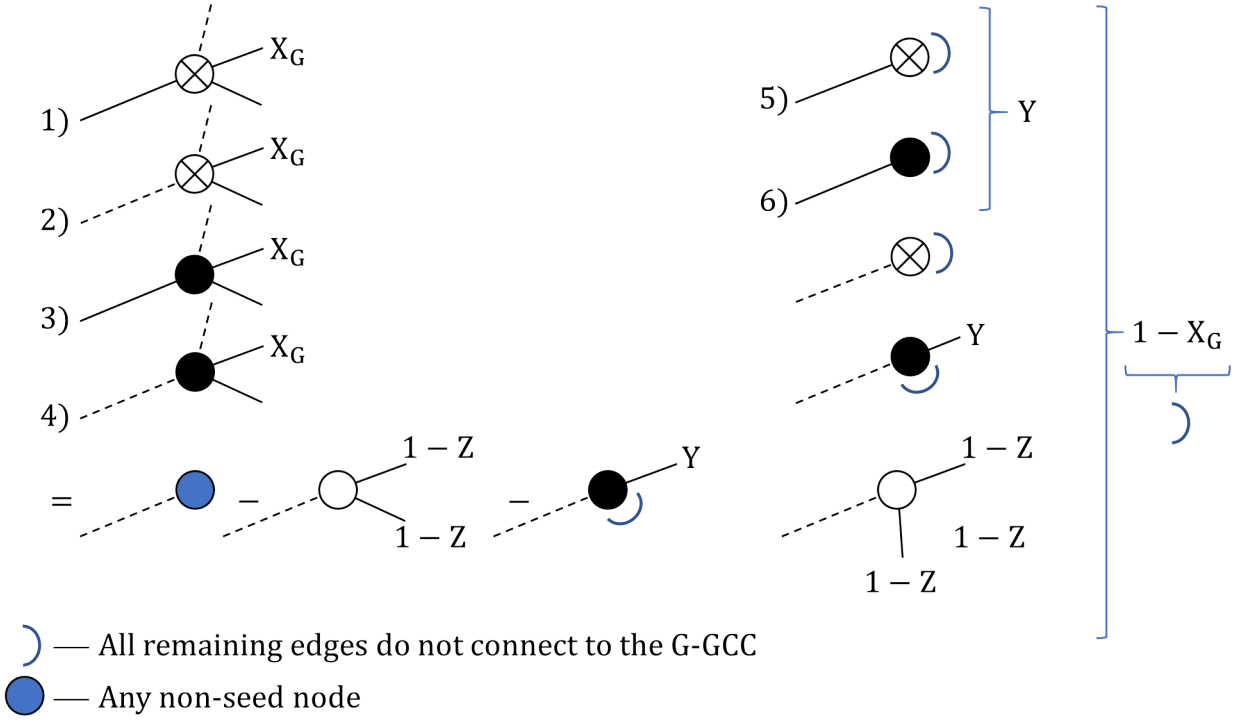


FIGURE 2.20: All the possible cases to connect to the G-GCC. To the previous image definitions, we add that the blue circle represents any non-seed node in the infinite time limit, whether it is recovered or not.

The cases of X_G are:

1 -The node we meet is a seed, the edge exists in the PRN, and that seed connects to the G-GCC. Notice that this is the only case considered in the Classic definition. Probabilistically

this term reads:

$$pf \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \sum_{l=1}^{q-1} \binom{q-1}{l} X_G^l (1 - X_G)^{q-1-l}; \tag{2.100}$$

2 -The node we meet is a seed, the edge was removed in the PRN, and that the seed connects to the G-GCC. Has we have seen since we are assuming that we are a recovered node, the existence (*usage* in transmitting the disease) or not of this edge should not make a difference in determining if we belong to the RGCC. This is one of the cases that is not counted in the Classic RGCC. Probabilistically this term reads:

$$(1-p)f \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \sum_{l=1}^{q-1} \binom{q-1}{l} X_G^l (1 - X_G)^{q-1-l}; \tag{2.101}$$

3 -The node we meet is not a seed, but the edge exists in the PRN, and the node connects via one of its edges (who is not me) to the G-GCC. One may ask, if we should not make sure if this node we meet is recovered as well, in order for him to be able to connect to G-GCC, and yes we should. But do notice that we are defining X_G as the probability that I connect to the G-GCC via this edge, given that I am recovered. Well, if I am recovered, and I connect to a non-seed node upon following an edge in the PRN, it means that the non-seed node necessarily got infected (and then recovered), at least having been passed the infection through me. So, since it is recovered, it just needs to connect to the G-GCC via any of the other edges. Probabilistically this term reads:

$$p(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \sum_{l=1}^{q-1} \binom{q-1}{l} X_G^l (1 - X_G)^{q-1-l}; \tag{2.102}$$

4 -The node we meet is not a seed, and this edge that I am following was removed in the PRN, but it got infected via someone else (that cannot be me as in the previous case, since this edge did not get *used*), and it connects to the G-GCC.

Notice that the fourth way of belonging to the G-GCC: by following an edge and meeting a non seed node, via an edge that was removed, is hard to construct mathematically since it is no longer a subset of Z, like X_C is.

The way we overcome this is by using negated cases, that is: we construct the possibilities (and probabilities) that we follow a removed/damaged edge, and meet a non-seed node who does *NOT* connect to the G-GCC, and then subtract them from one. The only possible cases in which this happens is if either the non-seed node we meet does not get infected ever (all its

edges are cases of $1 - Z$), or it got infected but by someone else that does not connect to the G-GCC. (This time, I cannot infect my non-seed neighbour, since the edge we are following was removed).

To do this, we introduce Y: the probability that if I am a non-seed node and follow one of my edges, that I get passed the infection but by someone who is not in the G-GCC. This can happen in two ways, they are:

5 - I connect to a seed, this edge exists in the percolated graph (in order for me to be passed the infection), but the seed does not connect to the G-GCC, through its other edges. Probabilistically this term reads:

$$pf \sum_{q=1}^{\infty} \frac{qP(q)}{< q >} (1 - X_G)^{q-1}; \tag{2.103}$$

6 - I connect to a non-seed, this edge exists in the PRN (again, to pass the infection to me), and this non-seed got infected by someone else that does not connect to the G-GCC (just like me). Probabilistically this term reads:

$$p(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \sum_{l=1}^{q-1} \binom{q-1}{l} Y^l (1 - X_G - Y)^{(q-1-l)}; \tag{2.104}$$

Notice from Figure 2.20, that these are $1 - X_G$ cases, making Y and X_G mutually exclusive..

Lastly take into account that this way of defining X_G and Y, we are including in the G-GCC the possibility that a cluster of recovered nodes connects via a damaged edge to another cluster of recovered nodes, that connects to the G-GCC via also a damaged edge, such as cluster C3 in presented Figure 2.21.

Since all the cases that constructed X_G and Y are independent of each other, we can simply sum them to obtain to obtain each of these quantities:

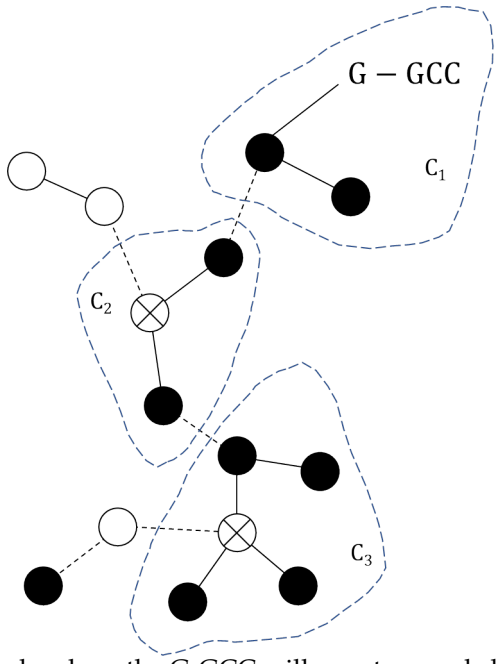


FIGURE 2.21: An example where the G-GCC will count several clusters connected via damaged edges, such as C2 and C3, when only C2 has a connection to the G-GCC

$$X_G = 1 + 2 + 3 + 4 \tag{2.105}$$
$$= pf \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \sum_{l=1}^{q-1} \binom{q-1}{l} X_G^l (1 - X_G)^{q-1-l}$$
$$+ (1-p) f \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \sum_{l=1}^{q-1} \binom{q-1}{l} X_G^l (1 - X_G)^{q-1-l}$$
$$+ p(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \sum_{l=1}^{q-1} \binom{q-1}{l} X_G^l (1 - X_G)^{q-1-l}$$
$$+ (1-p)(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \left[1 - (1-Z)^{q-1} - \sum_{l=1}^{q-1} \binom{q-1}{l} Y^l (1 - X_G - Y)^{(q-1-l)} \right]$$

$$Y = 5 + 6 \tag{2.106}$$
$$= pf \sum_{q=1}^{\infty} \frac{qP(q)}{< q >} (1 - X_G)^{q-1} + p(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \sum_{l=1}^{q-1} \binom{q-1}{l} Y^l (1 - X_G - Y)^{(q-1-l)}.$$

Each of the terms, corresponds in order to each of the presented cases.

We should make a note on term 4. Notice that we are calculating the probability that the following does not happen: I meet a non seed node and it does not get infected, plus the probability that it does get infected but the nodes that pass the infection to it do not connect to the G-GCC. The probability that a non-seed node does not get infected via one of its nodes is $(1 - Z)$, so the probability that it does not via any of its edges is $(1 - Z)^{q-1}$ (because the

remaining edge is connecting to me, which we know was removed). Regarding the second of these possibilities, let's say that upon following the edge I meet node A. For A to be passed the infection via a neighbour that will not connect to the G-GCC, node A's remaining edges must not connect to the G-GCC as well, otherwise, the whole branches from which A got passed the infection from, would connect to it. So unlike the first three terms, where there are no restrictions to the edges that do not meet the G-GCC, here these restrictions exist. We have to ensure that A's edges, from which it did not get passed the infection, do not connect to the G-GCC. This is done by the subtracting X_G, in the term $(1 - X_G - Y)^{q-1-l}$.

Before we move on, notice that unlike X_G which is not necessarily a subset of Z, Y is. Y represents the cases of Z (non-seed nodes that got passed the infection via at least one of their neighbours), that do not connect to the G-GCC. The existence of redundant edges in cases of Y does not matter as well, as in the example where a non-seed node gets passed the infection via two different paths, but both these paths do not connect to the G-GCC. Figure 2.22 shows an example of this.

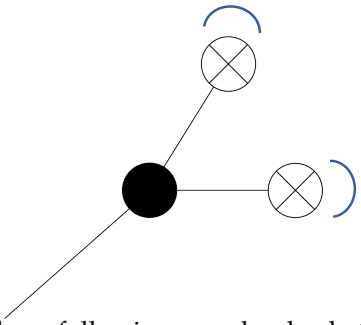


FIGURE 2.22: An example where following an edge leads to a node that got passed the infection by two seeds, implying that at least one of those edges is redundant. Nevertheless, this does not have an implication on the validity of Y, since none of those seeds connect to the G-GCC

These equations can be simplified using tricks like the ones we used for X_C and Z. For Y we have:

$$Y = pf \sum_{q=1}^{\infty} \frac{qP(q)}{< q >}(1 - X_G)^{q-1} + p(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{< q >}[(1 - X_G)^{q-1} - (1 - X_G - Y)^{(q-1)}], \tag{2.107}$$

Notice one more time, that we can start each summation in $q = 1$, making these summations all in the shapes we have seen before.

$$Y = pfe^{-\lambda X_G} + p(1-f)(e^{-\lambda X_G} - e^{-\lambda(X_G+Y)}) \tag{2.108}$$
$$Y = p[fe^{-\lambda X_G} + (1-f)e^{-\lambda X_G} - (1-f)e^{-\lambda(X_G+Y)}] \tag{2.109}$$
$$Y = p[e^{-\lambda X_G} - (1-f)e^{-\lambda(X_G+Y)}] \tag{2.110}$$
$$Y = pe^{-\lambda X_G}[1-(1-f)e^{-\lambda Y}]. \tag{2.111}$$

For X_G we have:

$$\begin{aligned} X_G = {} & pf\sum_{q=1}^{\infty}\frac{qP(q)}{<q>}[1-(1-X_G)^{q-1}] + (1-p)f\sum_{q=1}^{\infty}\frac{qP(q)}{<q>}[1-(1-X_G)^{q-1}] \\ & + p(1-f)\sum_{q=1}^{\infty}\frac{qP(q)}{<q>}[1-(1-X_G)^{q-1}] \\ & + (1-p)(1-f)\sum_{q=1}^{\infty}\frac{qP(q)}{<q>}\{1-(1-Z)^{q-1}-[(1-X_G)^{q-1}-(1-X_G-Y)^{(q-1)}]\}. \end{aligned} \tag{2.112}$$

Hence:

$$X_G = pf(1-e^{-\lambda X_G}) + (1-p)f(1-e^{-\lambda X_G}) + p(1-f)(1-e^{-\lambda X_G}) \tag{2.113}$$
$$+ (1-p)(1-f)(1-e^{-\lambda Z} - e^{-\lambda X_G} + e^{-\lambda(X_G+Y)})$$
$$= [f+p(1-f)](1-e^{-\lambda X_G}) + (1-p)(1-f)(1-e^{-\lambda Z} - e^{-\lambda X_G} + e^{-\lambda(X_G+Y)}) \tag{2.114}$$
$$= (1-e^{-\lambda X_G}) + (1-p)(1-f)(e^{-\lambda(X_G+Y)} - e^{-\lambda Z}) \tag{2.115}$$
$$= (1-e^{-\lambda X_G}) + (1-p)(1-f)(e^{-\lambda(X_G+Y)} - e^{-\lambda Z}) \tag{2.116}$$
$$= 1 - e^{-\lambda X_G} + (1-p)(1-f)e^{-\lambda(X_G+Y)} - (1-p)(1-f)e^{-\lambda Z} \tag{2.117}$$
$$= 1 - e^{-\lambda X_G} + (1-f)e^{-\lambda(X_G+Y)} - p(1-f)e^{-\lambda(X_G+Y)} - (1-f)e^{-\lambda Z} + p(1-f)e^{-\lambda Z}. \tag{2.118}$$

Noticing that we can write $\frac{Z}{p} = 1-(1-f)e^{-\lambda Z}$:

$$X_G = \frac{Z}{p} - e^{-\lambda X_G}[1-(1-f)e^{-\lambda Y}] - p(1-f)e^{-\lambda(X_G+Y)} + p(1-f)e^{-\lambda Z} \tag{2.119}$$

$$= \frac{Z}{p} - \frac{Y}{p} - p(1-f)e^{-\lambda(X_G+Y)} + p(1-f)e^{-\lambda Z} + pe^{-\lambda X_G} - pe^{-\lambda X_G} \tag{2.120}$$

$$= \frac{Z}{p} - \frac{Y}{p} + pe^{-\lambda X_G}(1-(1-f)e^{-\lambda Y}) + p(1-f)e^{-\lambda Z} - pe^{-\lambda X_G} \tag{2.121}$$

$$= \frac{Z-Y}{p} + Y + p(1-f)e^{-\lambda Z} - pe^{-\lambda X_G} + p - p \tag{2.122}$$

$$= \frac{Z-Y}{p} + Y - Z - pe^{-\lambda X_G} + p \tag{2.123}$$

$$= (Z-Y)\left(\frac{1}{p} - 1\right) + p(1-e^{-\lambda X_G}) \tag{2.124}$$

$$\Leftrightarrow X_G = (Z-Y)\left(\frac{1-p}{p}\right) + p(1-e^{-\lambda X_G}). \tag{2.125}$$

The way in which X_G and Y are written in Equation (2.125) and (2.111), respectively, is probably the simplest way to write them, and in future calculations we will use these forms of the equations since they are easier to deal with. However, it is possible to write an independent equation for each of them.

Notice that from Equation (2.111) we can write

$$e^{-\lambda X_G} = \frac{Y}{p[1-(1-f)e^{-\lambda Y}]} \tag{2.126}$$

$$\Leftrightarrow X_G = -\frac{1}{\lambda}\log\left(\frac{Y}{p[1-(1-f)e^{-\lambda Y}]}\right), \tag{2.127}$$

and substituting into Equation (2.125)

$$-\frac{1}{\lambda}\log\left(\frac{Y}{p[1-(1-f)e^{-\lambda Y}]}\right) = (Z-Y)\left(\frac{1-p}{p}\right) + p\left(1 - \frac{Y}{p[1-(1-f)e^{-\lambda Y}]}\right) \tag{2.128}$$

$$\Leftrightarrow Y = p[1-(1-f)e^{-\lambda Y}]e^{-\lambda\left\{(Z-Y)\left(\frac{1-p}{p}\right)+p\left[1-\frac{Y}{p\left(1-(1-f)e^{-\lambda Y}\right)}\right]\right\}}. \tag{2.129}$$

Obtaining a self-consistency equation just for Y. Using the same approach for X_G, from Equation (2.125):

$$Z - Y = \frac{p}{1-p}[X_G - p(1 - e^{-\lambda X_G})] \tag{2.130}$$

$$Y = Z - \frac{p}{1-p}[X_G - p(1 - e^{-\lambda X_G})]. \tag{2.131}$$

Substituting in Equation (2.111) in the left side of the equation:

$$Z - \frac{p}{1-p}[X_G - p(1 - e^{-\lambda X_G})] = pe^{-\lambda X_G}[1 - (1-f)e^{-\lambda Y}] \tag{2.132}$$

$$X_G - p(1 - e^{-\lambda X_G}) = -\frac{(1-p)}{p}\left\{-Z + pe^{-\lambda X_G}[1 - (1-f)e^{-\lambda Y}]\right\} \tag{2.133}$$

$$X_G = p(1-e^{-\lambda X_G}) + \frac{(1-p)}{p}Z - (1-p)e^{-\lambda X_G}[1-(1-f)e^{-\lambda Y}] \tag{2.134}$$

$$X_G = p + \frac{(1-p)}{p}Z - e^{-\lambda X_G}\left\{p + (1-p)[1 - (1-f)e^{-\lambda Y}]\right\}, \tag{2.135}$$

and finally:

$$X_G = p + \frac{(1-p)}{p}Z - e^{-\lambda X_G}\left[1 - (1-p)(1-f)e^{-\lambda\left\{Z - \frac{p}{1-p}\left[X_G - p(1-e^{-\lambda X_G})\right]\right\}}\right]. \tag{2.136}$$

This way we are able to write the self-consistency equations in an independent way. Notice that both X_G and Y depend on Z, however we can solve Equation (2.78) for Z first, and then calculate X_G and Y.

Now, in order for us to obtain the fraction of nodes that belong to the G-GCC, S_{X_G}, we have to remember that we defined X_G as coming from a recovered node. So we just need to see which nodes end up recovered and add the condition that they connect to the GCC:

$$S_{X_G} = f\sum_{q=1}^{\infty}P(q)\sum_{l=1}^{q}\binom{q}{l}X_G^l(1-X_G)^{q-l} + (1-f)\sum_{q=1}^{\infty}P(q)\left[1-(1-Z)^q - \sum_{l=1}^{q}\binom{q}{l}Y^l(1-X_G-Y)^{q-l}\right]. \tag{2.137}$$

The first term refers to the seeds, they mandatorily end up recovered, so they just need to connect at least once to X_G. The second term refers to the non-seeds, these ones will need to be infected during the disease spreading, added to the condition that they belong in the

G-GCC. Once again it is easier to construct the negated cases as X_G intersects with Z, but is not its subset. So, in order for a non-seed node to not belong in the G-GCC either it was never infected ($(1-Z)^q$), or if it was, it was passed the infection by a node that does not lead to the G-GCC, so via Y, and its other edges cannot connect to the G-GCC as well.

We can simplify the equations for S_{X_G} in the following way:

$$S_{X_G} = f\sum_{q=1}^{\infty} P(q)[1-(1-X_G)^q] + (1-f)\sum_{q=1}^{\infty} P(q)\left\{1-(1-Z)^q - [(1-X_G)^q - (1-X_G-Y)^q]\right\} \tag{2.138}$$

$$= f + (1-f)\sum_{q=1}^{\infty} P(q)[1-(1-Z)^q] - f\sum_{q=1}^{\infty} P(q)(1-X_G)^q \tag{2.139}$$

$$- (1-f)\sum_{q=1}^{\infty} P(q)\left[(1-X_G)^q - (1-X_G-Y)^q\right]$$

$$= S_Z - f\sum_{q=1}^{\infty} P(q)(1-X_G)^q - (1-f)\sum_{q=1}^{\infty} P(q)\left[(1-X_G)^q - (1-X_G-Y)^q\right]. \tag{2.140}$$

Notice that the total number of recovered nodes that belong to the G-GCC is the total number of recovered nodes, minus some of them represented by $f\sum_{q=1}^{\infty} P(q)(1-X_G)^q + (1-f)\sum_{q=1}^{\infty} P(q)\left[(1-X_G)^q - (1-X_G-Y)^q\right]$. Thinking about this term, it should represent the fraction of recovered nodes that do not belong to the G-GCC. In order to prove this, we will construct a quantity S_Y that represents exactly this. The recovered nodes that are not in the G-GCC are:

7- Seeds that do not connect to the G-GCC via any of their edges. Probablistically this term reads: $f\sum_{q=0}^{\infty} P(q)(1-X_G)^q$;

8- Non-seed nodes that got passed the infection via a neighbour that does not connect to the G-GCC, and neither do the rest of the non-seed nodes' edges. Probablistically this term reads: $(1-f)\sum_{q=1}^{\infty} P(q)\sum_{l=1}^{q}\binom{q}{l}Y^l(1-X_G-Y)^{q-l}$.

Since case 7 and 8 are mutually exclusive, we can write S_Y as:

$$S_Y = f\sum_{q=0}^{\infty} P(q)(1-X_G)^q + (1-f)\sum_{q=1}^{\infty} P(q)\sum_{l=1}^{q}\binom{q}{l}Y^l(1-X_G-Y)^{q-l} \tag{2.141}$$

$$= f\sum_{q=1}^{\infty} P(q)(1-X_G)^q + (1-f)\sum_{q=1}^{\infty} P(q)[(1-X_G)^q - (1-X_G-Y)^q]. \tag{2.142}$$

Which are exactly the terms that we subtract from S_Z in Equation (2.140). So we can write in general:

$$\Rightarrow S_{X_G} = S_Z - S_Y. \tag{2.143}$$

Making clear that all the total number of recovered nodes is $S_Z = S_{X_G} + S_Y$, the sum of the ones that belong to the G-GCC, and the recovered ones that do not.

Notice that this is not true for the equalities of X_G, Z and Y, because X_G is not a subset of Z (X_G includes cases of following damaged (not *used*) edges).

We can simplify the equations for S_Y using the same old recipe. For an Erdős-Rényi graph:

$$S_Y = f\sum_{q=0}^{\infty}\frac{e^{-\lambda}\lambda^q}{q!}(1-X_G)^q + (1-f)\sum_{q=1}^{\infty}\frac{e^{-\lambda}\lambda^q}{q!}(1-X_G)^q - (1-f)\sum_{q=1}^{\infty}\frac{e^{-\lambda}\lambda^q}{q!}(1-X_G-Y)^q. \tag{2.144}$$

Notice that all summations can start in $q = 0$:

$$S_Y = fe^{-\lambda X_G} + (1-f)e^{-\lambda X_G} - (1-f)e^{-\lambda(X_G+Y)} \tag{2.145}$$

$$\Leftrightarrow S_Y = e^{-\lambda X_G}\left[1-(1-f)e^{-\lambda Y}\right], \tag{2.146}$$

and

$$\Rightarrow S_{X_G} = S_Z - e^{-\lambda X_G}\left[1-(1-f)e^{-\lambda Y}\right]. \tag{2.147}$$

With these results, one can now solve the self-consistency equations for X_G and Y, and then substitute the solutions into Equation (2.147) obtaining the value of the fraction of the network that belongs to the G-GCC.

2.3.2 Limit of no G-GCC

Before we advance our analysis of S_{X_G}, let us show that in the limit that there is no G-GCC, Y meets Z. This is because the condition, in Y, that the node who passed the infection to me does not belong to the G-GCC is guaranteed. So, being myself a non-seed node, the probability that I get passed the infection via one of my edges by a node who does not belong in the G-GCC, Y, is simply the probability that I get passed the infection via one of my edges, Z, since no node belongs to the G-GCC.

Mathematically, for an Erdős-Rényi graph, if $X_G = 0$, then Y becomes (from Equation (2.125)):

$$0 = (Z - Y)\left(\frac{1-p}{p}\right) + p(1 - e^{-\lambda 0}) \tag{2.148}$$

$$\rightarrow Z = Y. \tag{2.149}$$

At the same time, from Equation (2.111):

$$Y = p[1 - (1-f)e^{-\lambda Y}]. \tag{2.150}$$

Which is is exactly like the self-consistency solution for Z, Equation (2.78).

However, we can prove this for any distribution $P(q)$:

$$Y = pf \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} + p(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \sum_{l=1}^{q-1} \binom{q-1}{l} Y^l (1-Y)^{(q-1-l)} \tag{2.151}$$

$$Y = pf + p(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} [1 - (1-Y)^{(q-1)}]. \tag{2.152}$$

Notice that it is the same self-consistency equation as Equation (2.66), which implies that $\Rightarrow Y = Z$ in this limit.

In order to check consistency, the equation for X_G should be zero on both sides. If we set $Y = Z$:

$$0 = (1-p)(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \left[1 - (1-Z)^{q-1} - \sum_{l=1}^{q-1} \binom{q-1}{l} Y^l (1-Y)^{(q-1-l)}\right] \tag{2.153}$$

$$0 = (1-p)(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \left\{1 - (1-Z)^{q-1} - [1 - (1-Y)^{q-1}]\right\} \tag{2.154}$$

$$0 = (1-p)(1-f) \sum_{q=2}^{\infty} \frac{qP(q)}{< q >} \left\{1 - (1-Z)^{q-1} - [1 - (1-Z)^{q-1}]\right\} = 0, \tag{2.155}$$

making the G-GCC consistent in this limit.

In what concerns S_Y we get from Equation (2.146), that in this regime:

$$S_Y = 1 - (1 - f)e^{-\lambda Y}. \tag{2.156}$$

As expected, since this is the equation for S_Z, it means that all the recovered nodes exist only in finite clusters, i.e. $S_Y = S_Z$, since $Y = Z$.

2.3.3 Determining X_G and Y

We will now present a way of determining X_G and Y. Even though we can write them as independent functions of each other, which we could solve using the same algorithm as for X_C, we will explain the way to solve numerically these two self-consistency equations in their dependent form, for the three following reasons:

- Firstly, for generality. Because, even though in the case of a Poisson distribution for the degree, we can write X_G and Y separately, this might not be true for other distributions;

-Secondly, because even though X_G and Y are coupled, it is actually analytically easier to deal with them in this form, and numerically faster than in the separate form;

-Thirdly, because in later chapters we will solve different pairs of coupled equations using this method.

The way we solved coupled self-consistency equations is an adaptation of our method of solving one variable self-consistency equations. Just like in the previous method, here we define LHS_G and RHS_G, as the functions of the left-hand side, and right-hand side of Equation (2.125), respectively. Then we also define LHS_Y and RHS_Y, as the functions of the left-hand side, and right-hand side of Equation (2.111), respectively.

Since, the right hand side functions depend on Z, for each fixed $\langle q \rangle$, we first calculated its value. Then, we set some initial conditions: LHS_{G_0} and LHS_{Y_0}, and calculated the right hand side functions using both these values ($RHS_{G_0} = RHS_G(LHS_{G_0}, LHS_{Y_0})$ and the same for RHS_Y). (Remember these are two variable functions, hence required the usage of both left sides). In the next iteration, we would attribute to the values each left hand side function, the value of the correspondent right hand side function obtained in the last iteration: $LHS_{G_1} = RHS_{G_0}$ and $LHS_{Y_1} = RHS_{Y_0}$. We would repeat this procedure until the difference in between the left hand side function, and the right hand side, of both equations, was smaller than a quality threshold ϵ: $(|LHS_{G_n} - RHS_{G_n}| \wedge |LHS_{Y_n} - RHS_{Y_n}|) < \epsilon$. This way we force the system to evolve until it finds a stable solution.

In this case, since we had access to the independent version of each function, we where able to determine that X_G had always a solution at $X_G = 0$ (as $Z = Y$ in this case) which was stable

before a critical point, $\langle q \rangle < \lambda_{critG}$, and unstable after it, coinciding with the appearance of a second (bigger) stable solution. Thus, the initial condition $LHS_{G_0} = 1$, would always meet the physically correct one. For Y, the condition $LHS_{Y_0} = 0$, also always meets the stable solution. However in general, one has to be careful not to set the initial conditions in a way that the stable solution is $LHS_{G_n} \vee LHS_{Y_n} = \infty$, or has unstable solutions.

Having obtained the values of X_G and Y one can now finally calculate S_{X_G}. Figure 2.23 plots some curves comparing the size of the G-GCC with the CRGCC.

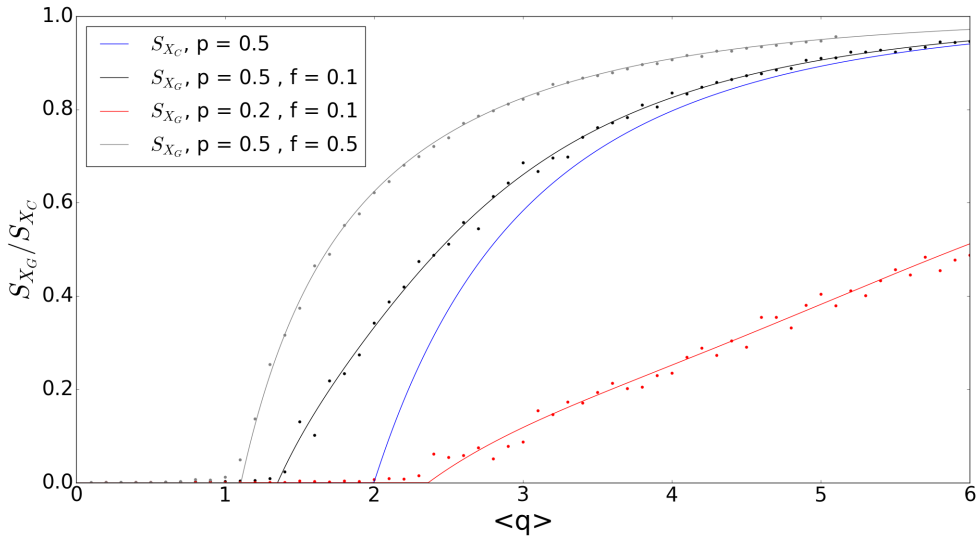


FIGURE 2.23: The evolution of the fraction of nodes belonging to each Giant Component with $\langle q \rangle$. Each line represents the theoretical prediction, for fixed p and f, and the dots of the same color, the numerical results obtained using the algorithm described in section 4.1. The numerical simulations were made on networks of size $N = 10,000$.

The first thing we can observe is that the numerical simulations coincide with the predicted theory for the G-GCC, reinforcing its validity over the CRGCC.

The second thing we can notice is that $S_{X_G} > S_{X_C}$, i.e the G-GCC is bigger than the CRGCC. This comes as no surprise since one of the reasons we proposed the G-GCC was because the CRGCC missed some nodes that connected to the RGCC in the Original Network. This difference is solely due to these cases, and as we can see they cannot be neglected. In fact, there is even a critical point, in which a G-GCC appears that occurs much before the appearance of the CRGCC.

Of course, as we increase $\langle q \rangle$ the G-GCC will grow, since each infected node has more edges to infect through. The fact that the S_{X_G} and S_{X_C} approximate for higher values of $\langle q \rangle$ is due to the fact that since there are a lot of connections for every node, the probability that it connects to the CRGCC increases. So, even if a node is recovered and has an edge to a recovered node in the CRGCC that got "incorrectly removed", the probability that this node connects to it

via somewhere else is very high, and the "incorrect removed" edge has no significance since the node will be included in the CRGCC anyhow.

Naturally, as p decreases, so does the G-GCC, for the same $\langle q \rangle$ and f. This happens for the same reason as for the CRGCC, that less edges exist in the PRN, corresponding to a decrease in the number of infected nodes, by each node. Also, smaller values of p are associated with higher recovery rates, and smaller infection rates.

Another interesting thing that we can observe is that the size of the G-GCC depends on the seed. As the seed grows, so does the G-GCC, and as it diminishes the G-GCC approaches the CRGCC. This is because, for higher values of the seed, more nodes will get infected, and so we will have more recovered nodes, in the GCC of the ON, who are neighbours but that did not get infected via each other.This does not interfere with the size of the CRGCC, because it is made out of nodes connected by undamaged edges in a Giant Component, in which there is at least a seed. So, the addition of seeds to the network only makes this Component to have more seeds, and not change its size. However, the addition of seeds does make more nodes (in finite clusters) end up recovered, increasing also the number of recovered clusters that connect to the CRGCC via damaged edges. These nodes, that in the Original Network are neighbours, are included in the G-GCC, making it dependent on f.

In fact, in the limit where $f = 1$ the G-GCC should meet the GCC. Due to the fact that all nodes end up necessarily recovered, and when the GCC appears, so will a G-GCC made of the same nodes. In the same line of thought, in the other limit, $f \to 0$, the G-GCC should meet the CRGCC. This is because in this limit (let's take the absolute limit in which there is only one seed), there won't be nodes who got passed the infection via different paths, since there are no different paths to be passed the infection from (there are no other seeds to infect; either a node got infected from the seed directly, or from someone who got infected from that same seed). Of course, since we are using an infinite network, and f represents a fraction of it, one can never say that there is only one seed. But the argument is just as valid, since there are so few seeds, that the probability that a node is passed the infection via two different branches (each branch leading to a different seed) goes to zero.

Finally, we believe that the G-GCC has never been noticed because most models are constructed and studied in the limit $f \to 0$ [32], and in this limit the G-GCC meets the CRGCC.

2.3.4 Limit $f \to 0$

In order to prove that the G-GCC meets the CRGCC, in this limit, we should first look at Z, and X_C:

$$Z = p[1 - (1 - f)e^{-\lambda Z}] \tag{2.157}$$

$$X_C = p(1 - e^{-\lambda X_C}) \tag{2.158}$$

and in the limit $f \to 0$:

$$Z = p(1 - e^{-\lambda Z}). \tag{2.159}$$

Notice than in this limit the equation for Z becomes exactly the same as the equation for X_C, and so do their solutions. Hence:

$$Z_0 = X_{CRGCC_0}, \tag{2.160}$$

where the index zero was introduced to remind us that this is only valid in this limit.

Notice that this also happens for S_Z and S_{X_C}:

$$S_Z = 1 - (1 - f)e^{-\lambda Z} \tag{2.161}$$

$$S_{X_C} = 1 - e^{-\lambda X_C}. \tag{2.162}$$

So the equation for S_Z will become:

$$S_{Z_0} = 1 - e^{-\lambda Z_0} = 1 - e^{-\lambda X_{CRGCC_0}} = S_{X_{CRGCC_0}}. \tag{2.163}$$

This is a very interesting result, since it means that there are so few seeds, that the behaviour of the system is completely dominated by the ones in the GCC. Also, it is the only case in which we also observe a second order phase transition in S_Z. This transition works exactly like the one for S_{X_C}.

Now, since $S_{X_G} \leq S_Z$, because the number of recovered who belong in the G-GCC is at most the total number of recovered nodes; and $S_{X_G} \geq S_{X_C}$, since the G-GCC includes all the cases that the CRGCC does, plus the discriminated cases; this means that $S_{X_C} \leq S_{X_G} \leq S_Z$. Thus, since when $f \to 0$, $S_{X_C} \to S_Z$, this implies: $\Rightarrow S_{X_G} \to S_{X_C}$.

To prove that $S_{X_G} \leq S_Z$, we just have to notice that $S_Y \in [0,1]$ since it is a fraction, hence S_Y is always (zero or) positive. As S_{X_G} is equal to S_Z minus a (zero or) positive quantity $\Rightarrow S_{X_G} \leq S_Z$.

The other limit of our proof that $S_{X_G} \to S_{X_C}$, when $f \to 0$, can also be proved in a mathematical way, other than the heuristic argument. A way to do this is by proving that in this limit, $S_Y \to 0$, and we get that $S_{X_G} = S_{Z_0} = S_{X_{CRGCC_0}}$.

A sufficient condition for $S_Y \to 0$ when $f \to 0$, is if in this limit Y goes to zero as well. By looking at Equation (2.111) we can see that when $f \to 0$, $Y = 0$ is always a solution. To prove that this solution is always stable in this limit, one just has to use Lyapunovs' rule [33], and check that the modulo of the derivative of the right hand side (RHS) of Equation (2.129) (the independent equation of Y), for Y at zero, is smaller than 1. This derivative is:

$$\frac{\partial RHS}{\partial Y} = pe^{f_1}\left\{\lambda(1-f)e^{-\lambda Y} + \left[1-(1-f)e^{-\lambda Y}\right]\frac{\partial f_1}{\partial Y}\right\}, \tag{2.164}$$

where we have defined f_1 as

$$f_1 = -\lambda\left\{(Z-Y)\left(\frac{1-p}{p}\right) + p\left(1 - \frac{Y}{p\left[1-(1-f)e^{-\lambda Y}\right]}\right)\right\}. \tag{2.165}$$

Calculating $\frac{\partial f_1}{\partial Y}$:

$$\frac{\partial f_1}{\partial Y} = \lambda\left\{\left(\frac{1-p}{p}\right) + p\left(\frac{1}{p\left[1-(1-f)e^{-\lambda Y}\right]} - \frac{Y[\lambda(1-f)e^{-\lambda Y}]}{p\left[1-(1-f)e^{-\lambda Y}\right]^2}\right)\right\}, \tag{2.166}$$

and substituting back:

$$\frac{\partial RHS}{\partial Y} = pe^{f_1}\left(\lambda(1-f)e^{-\lambda Y} + \lambda\left\{\left(\frac{1-p}{p}\right)[1-(1-f)e^{-\lambda Y}] + p\left(\frac{1}{p} - \frac{\lambda Y(1-f)e^{-\lambda Y}}{p[1-(1-f)e^{-\lambda Y}]}\right)\right\}\right). \tag{2.167}$$

Which at $Y = 0$ reads:

$$\frac{\partial RHS}{\partial Y}|_{Y=0} = pe^{f_1|_{Y=0}} \left\{ \lambda(1-f) + \lambda \left[\left(\frac{1-p}{p} \right) f + 1 - 0 \right] \right\} \tag{2.168}$$

$$\Leftrightarrow \frac{\partial RHS}{\partial Y}|_{Y=0} = pe^{-\lambda\left(Z\frac{1-p}{p}+p\right)} \left\{ \lambda(1-f) + \lambda \left[\left(\frac{1-p}{p} \right) f + 1 \right] \right\}. \tag{2.169}$$

And in the limit $f \longrightarrow 0$:

$$\frac{\partial RHS}{\partial Y}|_{Y=0} = pe^{-\lambda\left(Z_0\frac{1-p}{p}+p\right)}(\lambda + \lambda) \tag{2.170}$$

$$\Leftrightarrow \frac{\partial RHS}{\partial Y}|_{Y=0} = 2p\lambda e^{-\lambda\left(p+\frac{1-p}{p}Z_0\right)}. \tag{2.171}$$

To prove that this function is always in the interval $[0, 1]$, for every value of p and λ , we can write it in the form:

$$\frac{\partial RHS}{\partial Y}|_{Y=0} = 2\lambda' e^{-\lambda'} e^{-\lambda'\frac{1-p}{p^2}Z_0}. \tag{2.172}$$

Where $\lambda' = \lambda p$. Notice that the second exponential, $e^{-\lambda'\frac{1-p}{p^2}Z_0}$, is always smaller or equal to one since λ', Z_0, and $\frac{1-p}{p^2}$, are always positive or neutral.

Regarding the other exponential, it will have an extreme at:

$$\frac{\partial(\lambda' e^{-\lambda'})}{\partial\lambda'} = 0 \tag{2.173}$$

$$\Leftrightarrow e^{-\lambda'}(1-\lambda') = 0 \tag{2.174}$$

$$\Leftrightarrow \lambda' = 1, \tag{2.175}$$

and this extreme is a maximum since the second derivative at this point is negative:

$$\frac{\partial^2(\lambda' e^{-\lambda'})}{\partial\lambda'^2} = -e^{-\lambda'}(2-\lambda'), \tag{2.176}$$

which at $\lambda' = 1$, is:

$$\frac{\partial^2(\lambda' e^{-\lambda'})}{\partial\lambda'^2}|_{\lambda'=1} = -e^{-1} < 0, \tag{2.177}$$

so, the first exponential, from Equation (2.172), at it maximum is:

$$max(\lambda' e^{-\lambda'}) = e^{-1}. \tag{2.178}$$

As such, $\frac{\partial RHS}{\partial Y}|_{Y=0}$, will be a product of functions which are maximized by:

$$\frac{\partial RHS}{\partial Y}|_{Y=0} = 2 \times (\leq e^{-1}) \times (\leq 1) \Rightarrow (< 1) \times (\leq 1), \tag{2.179}$$

since $e > 2$. Which implies:

$$\Rightarrow \frac{\partial RHS}{\partial Y}|_{Y=0} < 1. \tag{2.180}$$

Now, since $\lambda' \in [0, \infty[$, and the exponentials are always positive, as well, in this interval. This implies that:

$$\Rightarrow \frac{\partial RHS}{\partial Y}|_{Y=0} \in [0, 1[. \tag{2.181}$$

As we wanted to demonstrate.

Having found a stable solution at $Y = 0$ in this limit, this implies that

$$S_{Y_0} = e^{-\lambda X_G}[1 - (1 - f)e^{-\lambda \times 0}] \tag{2.182}$$

$$\Rightarrow S_{Y_0} = f e^{-\lambda X_G} \longrightarrow 0, \tag{2.183}$$

implying also that

$$S_{X_{G_0}} = S_{Z_0} - S_{Y_0} \longrightarrow S_{Z_0}. \tag{2.184}$$

Since $S_{Z_0} \longrightarrow S_{X_{CRGCC_0}}$, thus $S_{X_{G_0}} \longrightarrow S_{X_{CRGCC_0}}$, as we wanted to demonstrate.

2.3.5 Critical Point for the Emergence of the G-GCC

As we have seen, just like in the CRGCC a critical point appears for the G-GCC. As the mean degree increases, the size of the recovered clusters will increase, since there will be more connections to get infected from. As we reach the critical point, λ_{critG}, the mean size of each cluster diverges, just like in the CRGCC case, and a cluster of infinite size made of recovered clusters appears, the G-GCC. Since we are allowing more connections between recovered clusters, the point in which some finite recovered clusters connect, making an infinite cluster, happens sooner (i.e. for smaller mean degree values). In our analogy of the fires, we are considering that the small fires connect more easily with each other, making bigger fires, than the CRGCC; thus, an infinte fire will appear sooner.

In order to calculate the critical point we can Taylor expand around $X_G = 0$, since we know that the G-GCC will appear from zero. We will use Equations (2.125) and (2.111), because it is actually easier to calculate the critical points using this equations rather than the independent version of them. However, one could also calculate the critical points using these equations by either Taylor expanding, or by noticing that the right side, RHS of (2.136) is monotonically increasing, and its derivative, monotonically decreasing to zero. So, when $\frac{\partial RHS}{\partial X_G} > 1$ a second solution appears.

We have:

$$X_G = (Z - Y)\left(\frac{1-p}{p}\right) + p(1 - e^{-\lambda X_G}), \tag{2.185}$$

and Taylor expanding around $X_G = 0$

$$X_G = (Z - Y)\left(\frac{1-p}{p}\right) + p\{1 - [1 - \lambda X_G + O(X_G^2)]\}. \tag{2.186}$$

Disregarding second order terms:

$$X_G = (Z - Y)\left(\frac{1-p}{p}\right) + p\lambda X_G. \tag{2.187}$$

Now, when there is no G-GCC, we know that $Y = Z$, so when the G-GCC just appears and is still very small, we can say that $Y = Z - \delta$. With δ being positive, because Y is a subset of Z; at this point some of the connections of Z already start to be included in X_G.

So

$$X_G = [Z - (Z - \delta)] \left(\frac{1-p}{p}\right) + p\lambda X_G \tag{2.188}$$

$$\Leftrightarrow X_G = \delta \left(\frac{1-p}{p}\right) + p\lambda X_G. \tag{2.189}$$

In what concerns the eq. of Y:

$$Y = pe^{-\lambda X_G}[1 - (1-f)e^{-\lambda Y}] \tag{2.190}$$

$$Z - \delta = pe^{-\lambda X_G}[1 - (1-f)e^{-\lambda(Z-\delta)}]. \tag{2.191}$$

Taylor expanding around $X_G = 0$ and $\delta = 0$ (which is also very small in this regime), and disregarding second order terms:

$$Z - \delta = p(1 - \lambda X_G)[1 - (1-f)e^{-\lambda Z}(1 + \lambda\delta)] \tag{2.192}$$

$$\Leftrightarrow Z - \delta = p(1 - \lambda X_G)[1 - (1-f)e^{-\lambda Z} - \lambda\delta(1-f)e^{-\lambda Z}] \tag{2.193}$$

$$= (1 - \lambda X_G)[Z - p\lambda\delta(1-f)e^{-\lambda Z}] \tag{2.194}$$

$$= (1 - \lambda X_G)\left[Z - p\lambda\delta\left(1 - \frac{Z}{p}\right)\right] \tag{2.195}$$

$$\Leftrightarrow Z - \delta = Z - p\lambda\delta + \lambda\delta Z - \lambda X_G Z + \lambda^2 X_G p\delta - \lambda^2 X_G Z\delta. \tag{2.196}$$

Notice that the terms $\lambda^2 X_G p\delta$ and $-\lambda^2 X_G Z\delta$ are of second order, so they can be disregarded.

$$\Rightarrow -\delta = -\lambda\delta(p - Z) - \lambda X_G Z \tag{2.197}$$

$$\delta = \frac{\lambda X_G Z}{1 - \lambda(p - Z)}. \tag{2.198}$$

Substituting δ in the equation for X_G:

$$X_G = -\frac{-\lambda X_G Z}{1 - \lambda(p - Z)}\left(\frac{1-p}{p}\right) + p\lambda X_G. \tag{2.199}$$

Giving the condition for the critical point:

$$0 = p\lambda_{critG} + \frac{\lambda_{critG} Z}{1 - \lambda_{critG}(p - Z)} \left(\frac{1-p}{p} \right) - 1. \tag{2.200}$$

Where we defined the value of λ in this limit as λ_{critG}, the critical point.

This equation is not analytically solvable, however with a bisection method we can obtain a numerical solution. We can use a bisection method, because the function on the right hand side of Equation (2.200), just has one zero, and is positive on one side of it, and negative on the other. Before we do that, notice that this is the same condition for the critical point of the CRGCC but with the extra term $\frac{\lambda_{critG} Z}{1-\lambda_{critG}(p-Z)}(\frac{1-p}{p})$, which is responsible for the advance of the critical point in the G-GCC. Notice also than in the limit $f \to 0$, $Z \to 0$ at the critical point, and this condition reads $\lambda_{critG} = \frac{1}{p}$, which is also the critical point of the CRGCC.

The bisection method was implemented in the following way: Firstly we would set two initial values for the mean degree, a and b, being a on the left of the zero, and b on the right. Because we know that the critical point can never be smaller than one (since that is the critical point for the GCC itself), we set $a = 0.9$; in what concerns b, we just set it as an extremely high value, so that it was for sure on the right of the zero (the value we used was $b = 999$). After this we would calculate the value of $F = p\lambda + \frac{\lambda Z}{1-\lambda(p-Z)}(\frac{1-p}{p}) - 1$ at a and at b, obtaining: F_a and F_b, respectively. In doing this one has to be careful, because the value of Z for a and b is different. So, before calculating F at a or b, one should calculate the value of Z at each point, and only then calculate F_a and F_b using the correct value of Z. With this, we would introduce a variable named $c = \frac{(a+b)}{2}$, the middle point of a and b, and calculated F_c (with the correspondent value of Z). Then, noticing the function is also continuous, one can use Bolzano's Theorem to check if there is a zero between c and b (or a). This can be done by seeing if the product $F_c \times F_b > 0$. If this condition is true, than it means that c and b are on the same side of the zero, with c being closer to it; so we would set $b = c$. Otherwise, if $F_c \times F_b < 0$, it would mean that the zero was still in the interval $[c, b]$, so in this case we would adapt the other extreme: $a = c$. Had we the luck of $F_c \times F_b = 0$, this would mean that c would be right at the critical point! In the following iteration, we would check the difference $|F_a - F_b|$, and repeat the procedure until this difference was smaller than a threshold ϵ. When this threshold would be reached, we would recalculate c, and define it has the critical point for this value of p and f.

2.4 Results for Limit $k = 1$

Having the ability to calculate the critical point, Figure 2.24 presents a graphic that shows how λ_{crit_G} changes with p, for different values of f. In this figure, the evolution of S_{X_G} with

$\langle q \rangle$, for a fixed f (a curve), corresponds to following a vertical line at a fixed p, with the critical point emerging when we cross the theoretical line corresponding to that value of f.

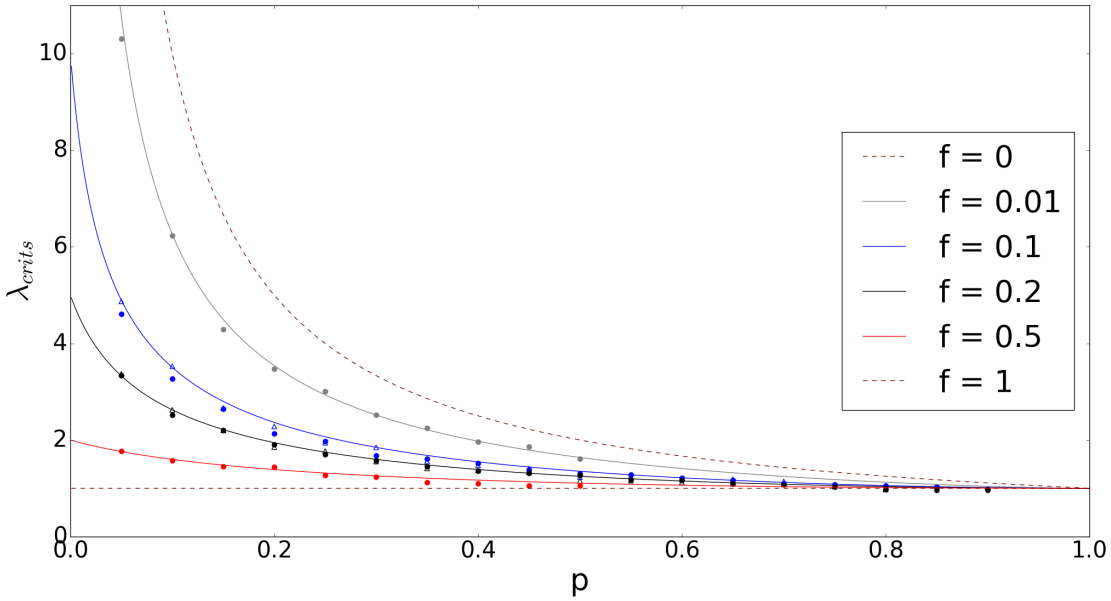


FIGURE 2.24: This graphic plots the value of $\langle q \rangle$ at which the G-GCC appeared for each curve, against p. The different values of f are represented by different colors. The lines represent the theoretical prediction for the size of the G-GCC at each value of f; and the dots represent the critical points we measured, for each curve, using the method described in section 4.2.1. The upper and lower dashed lines represent the theoretical prediction for the limits $f \to 0$ and $f \to 1$, respectively. The different shapes in dots of the same color, represent two different values of α and β that correspond to the same p. The value of α used in the red and gray dots was $\alpha = 0.5$, in the blue and black circular dots was $\alpha = 0.1$, and in the blue and black triangular dots was $\alpha = 0.2$. This way, the blue line represents the theoretical prediction for λ_{crit_G} at $f = 0.1$, the circular blue dots the critical points measured a system with $\alpha = 0.1$, and the triangular shaped dots the critical points measured in a system simulated with $\alpha = 0.2$. The size of the networks used for each curve was $N = 10,000$ nodes.

The first thing we can observe is that, for a fixed f, the curves are completely modeled by the parameter p, a ratio between α and β, just like in the CRGCC. This can be seen by noticing that the numerical data for the critical point coincides for different values of α, but same f. Once again, this means that what determines the impact of the disease is not α nor β per se, it is in a balance, p, between the two of them. In other words, if a disease is highly contagious having a very big value of β, it can have just as much of an impact in the population as a less contagious disease, if the ratio p is the same. Topologically, this is because, as explained before, since there are more edges kept in the PRN (which corresponds to a higher probability of an infected node passing the infection to someone before it recovers), one needs a smaller value of the mean degree for the recovered clusters to connect making an infinite component.

The second thing we can see, is the fact that, as expected, the critical points, for the same p, are different, for different seeds, f. What happens is that for higher values of f, and the same p, there will be a higher number of recovered nodes, who connect in the ON, but got

infected by different paths. In the CRGCC, these systems would all have a critical point in the same place, however, in the G-GCC since more clusters of recovered nodes are included, the critical points happen earlier. Since there are more finite Recovered clusters connecting, they make an infinite component sooner.

The dashed lines in Figure 2.24, represent the theoretical limits of $f = 1$, where everyone in the network starts infected, and $f \longrightarrow 0$, where the seed goes to zero.

In the first limit, as our intuition tells us, everyone ends up recovered, so the network will work just as a normal Erdős-Rényi graph, with critical point $\lambda_{crit_G} = 1$. Notice that the system is independent of p since it does not matter how contagious the disease is, nor its balance with the recovery probability α, since everyone will end up recovered anyhow. This happens even in the limit $p = 0$, where a disease is not transmittable ($\beta = 0$), or has an infinite recovery rate ($\alpha = 1$).

The other limit, $f \longrightarrow 0$, we have already seen that the curve for the G-GCC meets the one for the CRGCC, and so will the one for the critical point: $\lambda_{crit_G} = \lambda_{crit_{CRGCC}} = \frac{1}{p}$. Of course in reality, one can never have $f = 0$, because the disease would not exist in the first place, this is why we are representing this limit has $f \longrightarrow 0$. Another reason we believe the G-GCC is more complete, is because it also provides a solution to the system when the seed is not zero. So, if we are applying an SIR model, not to a disease, but to something that does not have necessarily a seed close to zero, the CRGCC fails to predict the impact of this contagion agent. An example of this, would be advertisement (of a series, restaurant,...), assuming f to be the number of people "infected" by the publicity when releasing the product. In the case of a series, β could be the probability that a person, when watching the series himself, that he "passed the infection" to his friends (neighbours) into watching it; and α the probability that the person finished watching the series (either because he saw all the episodes, or because he grew tired of it), case in which he would to stop talking about it and "recover".

For a non-zero seed, in the limit $p = 0$ the lines do not diverge, unlike the case for the CRGCC, because the seeds percolate in the ON forming a Giant Component, at the point $\langle q \rangle = \frac{1}{f}$. In the limit $p = 1$, the G-GCC meets the GCC as all the infected nodes pass the infection to their neighbours.

To finish we would like to comment that there is a global tendency for the numerical critical points to be slightly smaller than the actual value of λ_{crit_G}. This is mostly because of two things: the finite size effect, and numerical fluctuations.

The finite size effect, is given by the fact that the networks we are using are finite, and not infinite, as we are assuming. This has the consequence that the second order phase transition, that happens at the critical point, which in theory presents a discontinuity in the derivative of S_{X_G}, here does not. This is because, in a finite system, the relative size of the biggest

recovered connected component is never zero; i.e. there is no distinction between finite and infinite clusters. Any finite cluster will occupy a non-zero fraction of the network, and the critical point is estimated by looking for a sharp increase in the size of the largest cluster, as described in section 4.2.1. Below this point, the largest cluster is not zero, and grows as we get closer to the critical value. However, this effect can be diminished by using bigger networks, as indicated in [34]. Figure 2.25 compares two networks of different size, and the theory prediction, in which this effect can be observed.

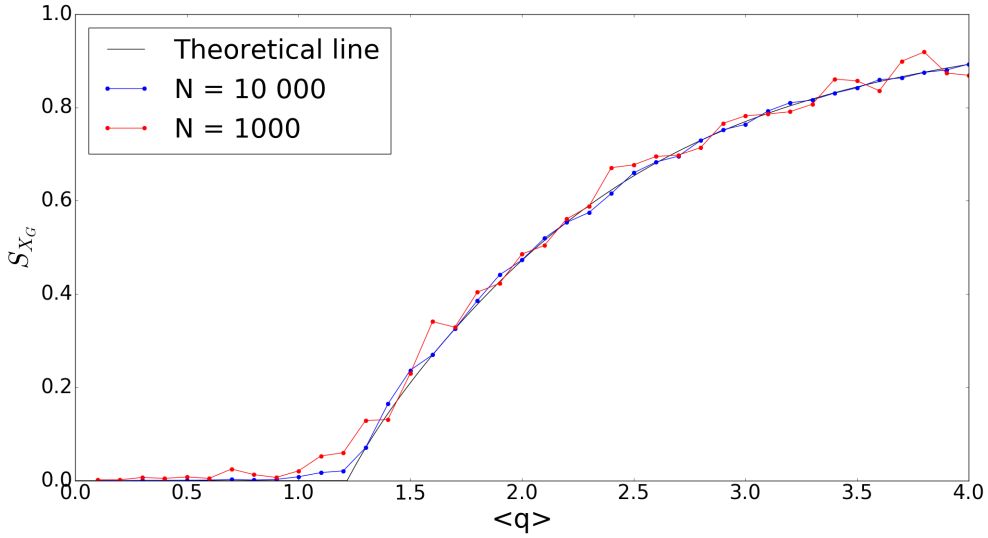


FIGURE 2.25: The figure shows the consequence of the finite size effect. The blue line was calculating using the algorithm presented in section 4.1, for a network of $N = 10,000$, and the red line for a network of $N = 1,000$. The parameters used were $p = 0.6$, $\alpha = 0.2$ and $f = 0.1$.

One can see that the derivative at the critical point of the graphic for $N = 10000$ is bigger than the one for $N = 1000$, and that the Theory Line has the biggest of them all. Thus, with our method of calculating the critical point, presented in section 4.2, the critical point can be "camouflaged" under the finite size effect. And some of the points still in the region where there is no G-GCC, would be included in the best fit line, decreasing its slope value, and the predicted value of λ_{crit_G}.

The effect of numerical fluctuations is given by the fact that the construction of the network is random. Since what we define is the probability that a node between two edges exists and the number of nodes N. Because of this, in practice, we could have a network with smaller mean degree λ_1, but actually with more edges than another with bigger mean degree λ_2, and/or more prone for a disease to spread. Of course, this effect is minimized as we increase N, since by the Law of Large Numbers, the degree distribution of the network will approach the theoretical one.

The second effect of numerical fluctuations is due to α and β being probabilities. As a consequence of this we could have nodes, that for the same number of neighbours, q, and a smaller value of β, infect more neighbours than other nodes (in a different network) with a higher value of β. Actually, we could even have everyone in the network recover, before any node got infected (other than the seeds), for any value of p; or the contrary that a single seed, for a very small β, started an infection that reached all nodes in the network. Once again, if we increase N, this effect is diminished, and the average of the number of neighbours each infected node of degree q passed the infection to, will meet pq.

Each of these effects, and much more their combination, can lead the network of a smaller mean degree, to behave as we would expect a network of higher mean degree to behave. Resulting in the fact that a G-GCC could actually appear in a network of $\langle q \rangle = \lambda_1$, but not in one of $\langle q \rangle = \lambda_2$, with $\lambda_2 > \lambda_1$.

In order to try to get around these problems, we sometimes used the theoretical line to guide us, and see if the critical point would be before or after which point; but still this was not perfect since sometimes the critical point would be deeply buried inside the numerical size effect, or a G-GCC would appear and disappear also camouflaging the critical point.

Notice that these effects are worse to determine critical points, when the slope of the derivative of the theoretical line is smaller. This is because, even if the finite size effect and the numerical fluctuations are small, they will already seem as part of the linear region after the critical point. Thus, they will be included in the linear regression, decreasing its slope. Figure 2.26 shows an example of this.

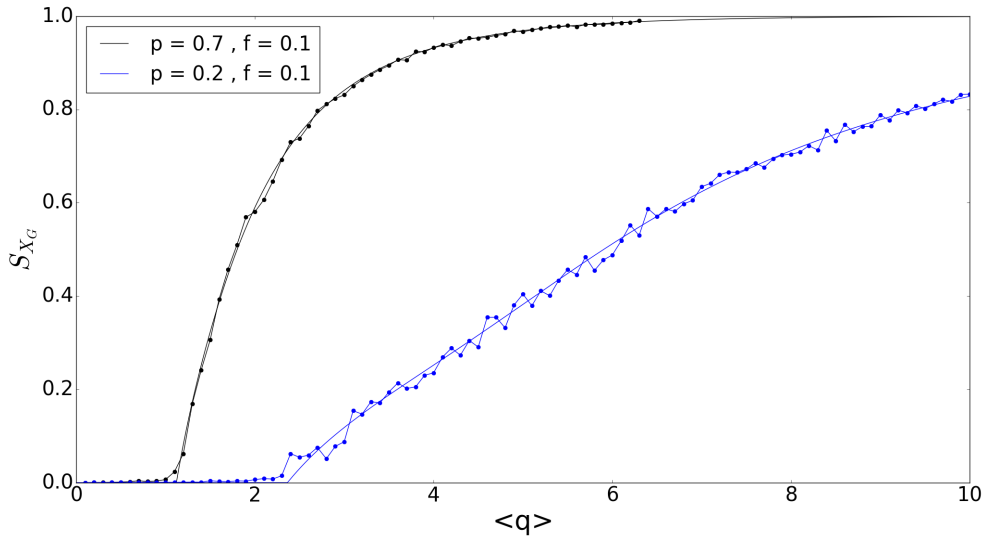


FIGURE 2.26: The figure shows the effect of fluctuations and the finite size effect. The lines represent the theoretical prediction, and the dots of the same color the results of the numerical simulation for the same parameters. The data was calculated using $\alpha = 0.1$ and $N = 10,000$.

Notice that these effects are also bigger in the curves where $\alpha = 0.1$, since these are the curves that correspond to the smallest values of β. In fact, the curves for $p = 0.05$ correspond to values of beta $\beta \lesssim 0.01$, which since it is such a small value, indicates that it is more sensitive to numerical fluctuations.

Another variable that is sensitive to numerical fluctuations is f. Smaller values of f are more prone to providing worse results than those with higher f. This is because if one seed recovers "before it should" (i.e. before passing the infection to the number of nodes predicted by the theory), it will correspond to a bigger decrease of the infected population (in percentage) for systems with smaller f, than for those with bigger. In the limit starting in a system with one seed, for a fixed p and $\langle q \rangle$ above the critical point, the probability that this seed recovers before it infects anyone (so not obtaining a G-GCC) is much bigger in this system than the probability of, in a similar system but with ten seeds, all the ten seeds recovering before they infected anyone. In addition, to the fact that the seeds may recover, it is also possible that the seeds are "not connected as they should", i.e. not distributed uniformly around the nodes of different degree (in theory, one could have all the seeds being isolated nodes). This is also less likely to happen in systems of higher f, due to the Law of Large Numbers.

Chapter 3

The Bootstrap Percolation Model

The second limit we will discuss is the limit when $\alpha \to 0$ and $\beta \neq 0$, where the SIR-Bootstrap Model meets the Bootstrap Percolation Model.

3.1 The Fraction of Recovered Nodes

As we have seen, it is of relevance to the understanding of the behaviour of the system to measure the fraction of the network that will end up active (infected) when the system stabilizes (i.e. in the infinite time limit).

In order to do this one can build a very similar argument to the one we did in the chapter 2. We can write the fraction of active nodes, that in our model will correspond to the recovered nodes, S_{Z_B}, as

$$S_{Z_B} = f + (1-f)\sum_{q=k}^{\infty} P(q) \sum_{l=k}^{q} \binom{q}{l} Z_B^l (1-Z_B)^{q-l}, \tag{3.1}$$

where Z_B was introduced as the probability that, being myself a node, I follow an edge and meet a node who is active in the infinite time limit, through other nodes who are not me.

So the terms that will belong to the active fraction of the network (for $t \longrightarrow \infty$) are: the seeds, given by the first term in the equation f; or any non-seed node who out of their q neighbours at least k of them are active in the infinite time limit, given by $(1-f)\sum_{q=k}^{\infty} P(q) \sum_{l=k}^{q} \binom{q}{l} Z^l (1-Z)^{q-l}$. Notice that the summation over l starts in $l = k$ as for a non-seed node to be active, it needs to have at least k active neighbours. This is also the reason why the summation over q starts in $q = k$.

The probability Z_B, is given by:

$$Z_B = f + (1-f)\sum_{q=k+1}^{\infty}\frac{qP(q)}{\langle q\rangle}\sum_{l=k}^{q-1}\binom{q-1}{l}Z_B^l(1-Z_B)^{q-1-l}. \tag{3.2}$$

Where the first term represents the probability that upon following an edge, being myself a node, I meet a seed, f. And second term represents the probability that I meet a non-seed node, $(1-f)$, of degree q, which in order to be active needs to have k active neighbours itself (other than me). The reason I cannot count as a node to activate my neighbour, is because Z_B would no longer be a sufficient condition to activate me. We would be making the fallacy of circular reasoning where I count to activate my neighbour and it would count to activate me. In the way it is written, we are tracking the activation steps back to the seeds, as we do in the case for Z in the SIR Model. Notice that the summation over q starts in $k+1$ because of this, my neighbour needs to have at least k other edges to be activated from and then still one to activate me.

We should point out that, just like in the previous chapter, we choose the nodes that are seeds in a uniform random way, i.e all nodes have the same probability of being a seed f.

An important note to make is that when we are saying that a node "activated" another, or contributed to the "activation" of another (such as Z_B, we are referring to a node that in the infinite time limit would become active, independent of the state of the node it is "activating". These nodes not necessarily being the ones, that during the course of the dynamics of the network, were active when the node their are "contributing to activate" became active.

Finally, a more general way one can study bootstrap percolation is on damaged network. Either on a site damaged network i.e. a network whose nodes where removed with probability $1-p$ before the starting of the process; or on bond damaged network, i.e. a network whose edges where removed with probability $1-p$ before the starting of the process. The work presented in [24] considers site damaged networks, however, we will present the equations for a bond damaged network, since we believe it is more relevant to this thesis. In fact, in [35], is presented the equivalence between bond and site percolation.

The equation for the active fraction of the network, S_{Z_p}, can be written in the same way as Equation (3.1), by changing $Z_B \to Z_{Boot_p}$, the probability that, being myself a node, I follow an edge through a not removed edge, and meet a node who is active in the infinite time limit, by other nodes who are not me. The equation for Z_{Boot_p} , can be written in the same way as S_{Z_B}, Equation (3.2), but adding the condition that the edge I am following did not get removed. Thus:

$$S_{Z_p}\left(Z_p\right) = S_{Z_B}\left(Z_B\right) \tag{3.3}$$

$$Z_p = p \times Z_B. \tag{3.4}$$

Because this mapping is simple, we will start by focusing on non-damaged networks ($p = 1$). We can simplify Equations (3.1) and (3.2) using similar steps as we have used in previous chapters:

$$Z_B = f + (1-f)\sum_{q=k+1}^{\infty}\frac{qP(q)}{\langle q\rangle}\sum_{l=k}^{q-1}\binom{q-1}{l}Z_B^l(1-Z_B)^{q-1-l} \tag{3.5}$$

$$= f + (1-f)\sum_{q=k+1}^{\infty}\sum_{l=k}^{q-1}\frac{qP(q)}{\langle q\rangle}\binom{q-1}{l}Z_B^l(1-Z_B)^{q-1-l} \tag{3.6}$$

$$= f + (1-f)\sum_{l=q}^{\infty}\sum_{q=l+1}^{\infty}\frac{qP(q)}{\langle q\rangle}\binom{q-1}{l}Z_B^l(1-Z_B)^{q-1-l}. \tag{3.7}$$

We can switch the order of the summations by noticing that

$$\sum_{q=k+1}^{\infty}\sum_{l=k}^{q-1}F(q,l) = \sum_{l=q}^{\infty}\sum_{q=l+1}^{\infty}F(q,l), \tag{3.8}$$

with $F(q,l)$ being any function of variables q and l. Figure 3.1, shows this mapping, in which it is clear that this switch of order of summation corresponds to the addition of the same terms.

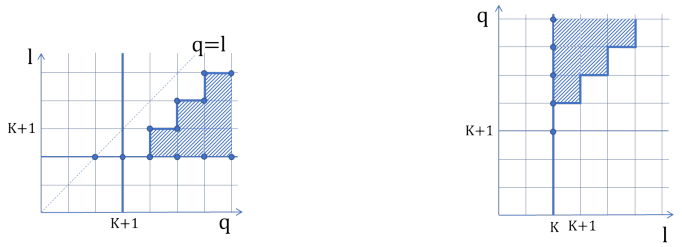


FIGURE 3.1: The figure shows the equivalency of the summations when switching the order of the terms. The dark blue line, dots and region represents the terms included in each summation.

Assuming we are working on an Erdős-Rényi graph

$$Z_B = f + (1-f)\sum_{l=k}^{\infty}\sum_{q=l+1}^{\infty}\frac{qe^{-\lambda}\lambda^q}{\langle q\rangle q!}\frac{(q-1)!}{l!\,(q-1-l)}(1-Z_B)^{q-1-l} \tag{3.9}$$

$$= f + (1-f)\sum_{l=k}^{\infty}\frac{Z_B^l\lambda^l}{l!}\sum_{q=l+1}^{\infty}\frac{e^{-\lambda}\lambda^{q-1-l}}{(q-1-l)!}(1-Z_B)^{q-1-l} \tag{3.10}$$

$$= f + (1-f)\sum_{l=k}^{\infty}\frac{(Z_B\lambda)^l}{l!}e^{-\lambda Z_B}\sum_{q=l+1}^{\infty}\frac{e^{-\lambda(1-Z_B)}\left[\lambda\left(1-Z_B\right)\right]^{q-1-l}}{(q-1-l)!}. \tag{3.11}$$

And by performing the change of variable $q' = q-1-l$, we get the summation over all terms of a Poisson distribution:

$$Z_B = f + (1-f)\sum_{l=k}^{\infty}\frac{(Z_B\lambda)^l}{l!}e^{-\lambda Z_B}\sum_{q'=0}^{\infty}\frac{e^{-\lambda(1-Z_B)}\left[\lambda\left(1-Z_B\right)\right]^{q'}}{q'!} \tag{3.12}$$

$$= f + (1-f)\sum_{l=k}^{\infty}\frac{(Z_B\lambda)^l}{l!}e^{-\lambda Z_B}. \tag{3.13}$$

Notice that the summation on the right side of Equation (3.13) is also a Poisson distribution, that is missing its first $k-1$ terms. So

$$Z_B = f + (1-f)\left(1 - \sum_{l=0}^{k-1}\frac{(Z_B\lambda)^l}{l!}e^{-\lambda Z_B}\right) \tag{3.14}$$

$$\Leftrightarrow Z_B = 1 - (1-f)\,e^{-\lambda Z_B}\sum_{l=0}^{k-1}\frac{(Z_B\lambda)^l}{l!} \tag{3.15}$$

$$\Rightarrow Z_p = p\left[1 - (1-f)\,e^{-\lambda Z_B}\sum_{l=0}^{k-1}\frac{(Z_B\lambda)^l}{l!}\right]. \tag{3.16}$$

Using a similar procedure for S_{Z_B}:

$$S_{Z_B} = f + (1-f)\sum_{q=k}^{\infty}P(q)\sum_{l=k}^{q}\binom{q}{l}Z_B^l\left(1-Z_B\right)^{q-l} \tag{3.17}$$

$$= f + (1-f)\sum_{l=k}^{\infty}\sum_{q=l}^{\infty}P(q)\binom{q}{l}Z_B^l\left(1-Z_B\right)^{q-l} \tag{3.18}$$

$$= f + (1-f)\sum_{l=k}^{\infty}\frac{(Z_B\lambda)^l}{l!}e^{-\lambda Z_B}\sum_{q=l}^{\infty}\frac{\left[\lambda\left(1-Z_B\right)\right]^{q-l}e^{-\lambda(1-Z_B)}}{(q-l)!} \tag{3.19}$$

$$S_{Z_B} = 1 - (1-f)\sum_{l=0}^{k-1}\frac{(Z_B\lambda)^l}{l!}e^{-\lambda Z_B}. \tag{3.20}$$

In order to solve the self consistency equation for Z_B one can use the algorithm used for Z presented in section 2.1. However, one needs to be careful, because whilst in the case for $k = 1$ only one stable solution appears in the interval $Z_B \in [0, 1]$, for $k > 1$ a second stable solution appears at $\lambda = \lambda_{crit2}$, with an unstable solution also appearing in between the two stable ones.

In Figure 3.2 we plot several curves for Z_B in which these solutions can be observed. Notice that for small λ only one solution exists, and it is only after $\lambda > \lambda_{transition}$ that a second (and third) solution appear.

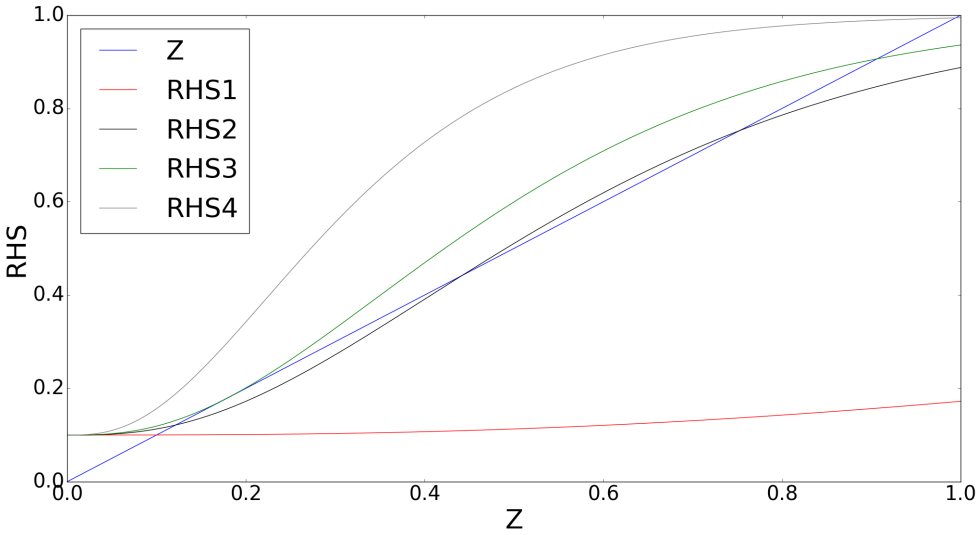


FIGURE 3.2: The figure shows the evolution of the function of the right hand side of Equation (3.20), for different values of mean degree, and at $k = 3$ and $f = 0.1$. $RHS1$ represents the line at $\langle q \rangle = 2$, where there is only one solution; $RHS2$ at $\langle q \rangle = 5 > \lambda_{transition}$, when there are three solutions; $RHS3$ at $\langle q \rangle = \lambda_{crit2}$, when the other two solutions disappear; and $RHS3$ at $\langle q \rangle = 9$, when there is only one solution again.

The appearance of the second stable solution is associated with the divergence of the mean size of subcritical clusters in the network. A subcritical cluster is a set of connected nodes that in the infinite time limit has exactly $k - 1$ active neighbours. Notice that if a single neighbour of one of the nodes in the subcritical cluster is activated, all the nodes in the critical cluster will be. This happens since the node whose neighbour was activated will become active itself, and activate all its neighbours who are also part of the subctritical cluster, as a consequence these neighbours will activate their neighbours who are in the subcritical cluster as well, and so on. This creates a chain event, called an avalanche, that ends with all the subcritical nodes in that cluster active [24]. Note that everytime we increase $\langle q \rangle$, i.e. "adding edges", we connect some of the subcritical clusters to active nodes, however the resulting increase in S_{Z_B} is limited when the subcritical clusters are finite. At the same time, we also increase the size of the subcritical clusters, until reaching a point that they connect into a Giant cluster, making the resulting avalanche infinite (the mean size of the subcritical

clusters diverges at this point). This creates a second solution in Z_B, and a discontinuous jump in S_{Z_B}. This phenomena is described in detail in [24].

In order to obtain the stable solution with physical meaning, we should start our algorithm in $Z = 0$, obtaining the smallest of them both. This is because of the way we build out problem: we are starting with a fraction f of active nodes, that later activate their neighbours, obtaining a growing fraction of activated nodes with time. So if we were to compute an evolution of Z_B with time, Z_t, it would reach the smallest stable solution. In order for the other solution to be the one with physical meaning, we would have to do a "reverse Bootstrap Model" in which all the nodes would start active, with a fraction of it starting deactivated. And the system would evolve as activated nodes would deactivate if they had less than a certain number of k active neighbours. Such a process is called "K-core Percolation".

Numerically what will happens, as we increase $\langle q \rangle$, is dependent on how we generate the graphs: if start with an Erdős-Rényi graph of parameters $\langle q \rangle$, N, and everytime we increase $\langle q \rangle$ we just add, to the same graph, the missing edges that are needed to reach its new value, we will obtain a curve for S_{Z_B} that increases monotonically, and has a jump (close to λ_{crit2} due to numerical fluctuations), that then remains "on top of the jump". However, if everytime we set a new $\langle q \rangle$, we generate a new graph from scratch, i.e. check the existence of each edge with probability r, it is possible that we get a graph, with $\langle q \rangle_1$, in which there was already a jump, and another with $\langle q \rangle_2 > \langle q \rangle_1$, in which there was not, due to numerical fluctuations. Graphically this will make the curve for S_{Z_B}, near λ_{crit2}, fluctuate a lot.

In our simulations, we opted for the latter, since it minimizes the chance of getting outlier results. This is because if we would generate a graph for λ_1, that would have an unrepresentative structure (such has not having a local tree like shape, most of the seeds being isolated,...), and add edges to this graph as we increased λ, this structure would interfere with the results for every $\langle q \rangle$ (even though its effects could, probabilistically, be minimized everytime we would add edges). On the other hand, if we generate a new graph for every $\langle q \rangle$, sometimes we could get a graph with a poor structure, but sometimes we would not. And the probability of generating graphs with poor structure through the whole range of $\langle q \rangle$ would be minimized. Also no systematic effects would be carried from one graph to the other, since each graph would be generated independently.

Figure 3.3 shows the evolution of S_{Z_B}, for some values of f and k.

Naturally S_{Z_B} increases with λ. This is because the system evolves, in a first step, when a node has k connections to seeds, and becomes active itself, increasing the active population. Consequently, this may make other nodes have k active neighbours, becoming active as well. And the system evolves in this way until there are no more inactive nodes with k active neighbours. With the increase of λ since there are a greater number of edges, not only does

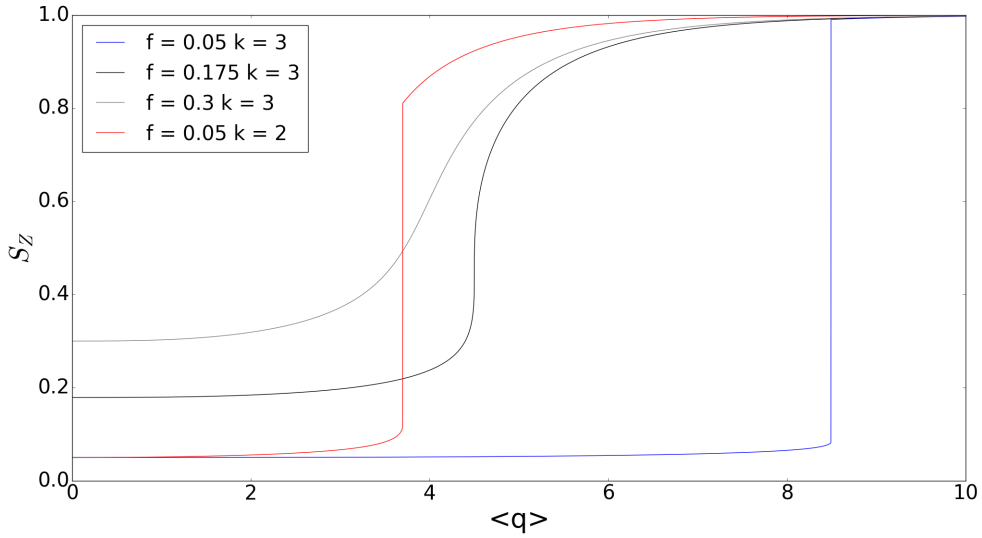


FIGURE 3.3: The evolution of S_{Z_B} with $\langle q \rangle$, for different values of f and k.

the probability that each node connects to k seeds increase, but also the probability that each node connects to someone who will become active eventually, leading to an increase in S_{Z_B}.

With the increase of f, notice that the whole curve seems to move to the left. This is for a similar reason: that since there are more seeds, the probability that each node connects to k of them increases; consequently the probability that any node connects to someone who will get, eventually, active, increases as well.

Notice that on the appearance of the second stable solution of Z_B, S_{Z_B} suffers a jump. This corresponds to the inclusion of all the subcritical clusters that are now active. The point in which this happens, λ_{crit2}, is a critical point of a, commonly defined, hybrid phase transition [24]. It is called so because it has characteristics of both first and second order phase transitions. For starters, the solution for S_{Z_B} appears in a discontinuous way, giving the shape of a jump, typically associated with first order phase transitions. However, this point is associated with the divergence of the mean size of the subcritical clusters, which in a way can be seen as a susceptibility, associated with second order phase transitions.

Looking at the evolution of this critical point with f, we can see that the jump becomes smaller and smaller, as f grows. In fact, there is a value of f in which the jump disappears, corresponding to a continuous evolution of the solution of Z_B. We will define this value as f_{crit}. The reason the jump shrinks, is because more nodes are active from the start, so, for $\lambda < \lambda_{crit2}$, the probability that there is at least one node who is selected to be a seed in a otherwise finite subcritical cluster, grows. When this happens the whole subcritical cluster gets activated, meaning that the fraction of subcritical clusters that will "survive" without the introduction of a seed, for fixed $\langle q \rangle$, will grow less and less, making them in total represent a smaller fraction of the network. In fact, at f_{crit}, there are so many seeds, that the subcritical

clusters can never grow to an infinite size without one of its nodes being selected as a seed first, making the hybrid phase transition disappear.

The increase of k moves the curve to the right. This is because for the criteria for getting active becomes "harder", i.e. for the same $\langle q \rangle$ and f, each node will have to connect to more active neighbours. So, the same value of S_{Z_B}, is only obtained for higher values of mean degree, or higher values of f.

3.2 The Bootstrap Giant Connected Component

Just like in the SIR Model, in the Bootstrap Model a Giant Connected Component (i.e. a finite fraction of the infinite network), made entirely out of active (Recovered) nodes, appears. We will define it has Bootstrap Giant Connected Component, or BGCC.

In order to construct the fraction of the network that belongs to the BGCC, we will start by discussing X_B, the probability that, being myself a node, I follow an edge, in the infinite time limit, and meet a node that is active and connects to the BGCC via one of its neighbours (other than me).

In order to construct X_B, one is tempted to write it has we have been doing so far:

$$X_B = f \sum_{q=2}^{\infty} \frac{qP(q)}{\langle q \rangle} \sum_{m=1}^{q} \binom{q}{m} X_B^m \left(1 - X_B\right)^{q-m} \tag{3.21}$$

$$+ (1-f) \sum_{q=k+1}^{\infty} \frac{qP(q)}{\langle q \rangle} \sum_{l=k}^{q} \binom{q}{l} \left(1 - Z_B\right)^{q-l} \sum_{m=1}^{l} \binom{l}{m} X_B^m \left(Z_B - X_B\right)^{q-m}, \tag{3.22}$$

with the first term representing the probability that, being myself a node, I follow an edge and meet a seed who connects to the BGCC; and the second term representing the probability that I meet a non-seed node, who became active by having k or more active neighbours, other than me, and of those active neighbours, at least one connects to the BGCC.

However, this way of writing misses the case in which I connect to the BGCC via a node that I contribute to activate. Remember that Z_B includes only the cases that contributed to my activation. So, the way previously presented $X_B \subset Z_B$, means I can only connect to the BGCC through a node who activated me. But, if one of my neighbours was inactive with $k-1$ active neighbours, and due to my activation it becomes active and connects to the BGCC via its other-than-me neighbours, then I will connect to the BGCC through it.

In order to resolve this we need to divide X_B in two parts: X_a the probability that I follow an edge, in the infinite time limit, and meet an active node, which is active regardless of my

state, that is part of the BGCC; and X_b the probability that I follow an edge, in the infinite time limit, and meet an active node that is part of the BGCC, which only became active because I was active.

Notice that $X_a \subset Z_B$, and $X_b \subset 1 - Z_B$.

These two quantities satisfy the following equations:

$$\begin{aligned} Z_B - X_a =& f \sum_{q=2}^{\infty} \frac{qP(q)}{\langle q \rangle} (1 - X_B)^{q-1} \\ &+ (1-f) \sum_{q=k+1}^{\infty} \frac{qP(q)}{\langle q \rangle} \sum_{l=k}^{q-1} \binom{q-1}{l} (Z_B - X_a)^l (1 - Z_B - X_b)^{q-1-l} \end{aligned} \tag{3.23}$$

$$X_b = (1-f) \sum_{q=k}^{\infty} \frac{qP(q)}{\langle q \rangle} \binom{q-1}{k-1} \left[Z_B^{k-1} (1 - Z_B)^{q-k} - (Z_B - X_a)^{k-1} (1 - Z_B - X_b)^{q-k} \right], \tag{3.24}$$

where $X_B = X_a + X_b$.

The first equation represents the probability that, being myself a node, I follow an edge and meet a node who is active in the infinite time limit, but does not connect to the BGCC. For this to happen I can meet a seed, f, of any degree, $\frac{qP(q)}{\langle q \rangle}$, that does not connect to the BGCC via any of its edges, $(1 - X_B)^{q-1}$. Or, I can meet a non-seed node, $1 - f$, which became active via neighbours that also does not connect to the BGCC, $(Z_B - X_a)^l$ (notice that $l \geq k$); and of its other neighbours, none of them connects to the BGCC neither, $(1 - Z_B - X_b)^{q-1-l}$. Notice that the summation over q starts in $q = k + 1$ since the node I meet needs one edge to connect to me and at least k more to be active.

The second Equation, (3.24), accounts the possibility of following an edge (being myself active) and meeting a node that had $k - 1$ other neighbours which where active. In this case, the node I meet only became active because of me (i.e. if I had not become active, neither would these nodes. There would not be another node who eventually would become active and activate it since this is already looking in the infinite time limit). Nodes in these conditions are non-seed nodes, $1 - f$, of degree at least $q = k$, who have $k - 1$ other than me active neighbours, Z^{k-1}, that connect to X_a or X_b. This can be written probablistically as all the possiblilities of connecting to $k - 1$ active neighbours, and subtracting of these the ones that do not connect to the BGCC in any way: $\binom{q-1}{k-1} \left[Z_B^{k-1} (1 - Z_B)^{q-k} - (Z_B - X_a)^{k-1} (1 - Z_B - X_b)^{q-k} \right]$.

Notice that X_a and X_b are mutually exclusive.

We can simplify the equations for X_a and X_b using a similar approach as for Z_B:

$$Z_B - X_a = f \sum_{q=1}^{\infty} \frac{e^{-\lambda}\lambda^{(q-1)}}{(q-1)!} (1 - X_a)^{(q-1)} + (1-f) \sum_{l=k}^{\infty} \sum_{q=l+1}^{\infty} \frac{qP(q)}{\langle q \rangle} \binom{q-1}{l} (Z_B - X_a)^l (1 - Z_B - X_b)^{(q-1-l)} \tag{3.25}$$

$$Z_B - X_a = fe^{-\lambda X} + (1-f) \sum_{l=k}^{\infty} \frac{[\lambda (Z_B - X_a)]^l}{l!} \sum_{q=l+1}^{\infty} \frac{e^{-\lambda}\lambda^{(q-1-l)}}{(q-1-l)!} (1 - Z_B - X_b)^{(q-1-l)} \tag{3.26}$$

$$Z_B - X_a = fe^{-\lambda X} + (1-f) \sum_{l=k}^{\infty} \frac{[\lambda (Z_B - X_a)]^l}{l!} e^{-\lambda(Z_B + X_b)} \sum_{q=l+1}^{\infty} \frac{e^{-\lambda(1-Z_B-X_b)} [\lambda (1 - Z_B - X_b)]^{(q-1-l)}}{(q-1-l)!}, \tag{3.27}$$

and with a change of variable $q' = q - 1 - l$, we get that

$$\sum_{q=l+1}^{\infty} \frac{e^{-\lambda(1-Z_B-X_b)} [\lambda (1 - Z_B - X_b)]^{(q-1-l)}}{(q-1-l)!} = \sum_{q=0}^{\infty} \frac{e^{-\lambda(1-Z_B-X_b)} [\lambda (1 - Z_B - X_b)]^{(q')}}{(q')!} = 1. \tag{3.28}$$

So:

$$Z_B - X_a = fe^{-\lambda X} + (1-f) \sum_{l=k}^{\infty} \frac{[\lambda (Z_B - X_a)]^l}{l!} e^{-\lambda(Z_B + X_b)} \tag{3.29}$$

$$= fe^{-\lambda X} + (1-f) e^{-\lambda(X_a + X_b)} \sum_{l=k}^{\infty} \frac{[\lambda (Z_B - X_a)]^l}{l!} e^{-\lambda(Z_B - X_a)} \tag{3.30}$$

$$= fe^{-\lambda X} + (1-f) e^{-\lambda(X_a + X_b)} \left\{ 1 - e^{-\lambda(Z_B - X_a)} \sum_{l=0}^{k-1} [\lambda (Z_B - X_a)]^l \right\} \tag{3.31}$$

$$= fe^{-\lambda X} + (1-f) e^{-\lambda(X_a + X_b)} \left\{ 1 - e^{-\lambda(Z_B - X_a)} \sum_{l=0}^{k-1} [\lambda (Z_B - X_a)]^l \right\} \tag{3.32}$$

$$\Leftrightarrow X_a = Z_B - e^{-\lambda X} + (1-f) e^{-\lambda(Z_B + X_b)} \sum_{l=0}^{k-1} [\lambda (Z_B - X_a)]^l . \tag{3.33}$$

Doing the same for X_b:

$$X_b = (1-f)\sum_{q=k}^{\infty}\frac{qP(q)}{\langle q\rangle}\binom{q-1}{k-1}\left[Z_B^{k-1}\left(1-Z_B\right)^{q-k}-\left(Z_B-X_a\right)^{k-1}\left(1-Z_B-X_b\right)^{q-k}\right] \tag{3.34}$$

$$= \frac{(1-f)\,\lambda^{k-1}}{(k-1)!}\sum_{q=k}^{\infty}\frac{e^{-\lambda}\lambda^{q-k}}{(q-k)!}\left[Z_B^{k-1}\left(1-Z_B\right)^{q-k}-\left(Z_B-X_a\right)^{k-1}\left(1-Z_B-X_b\right)^{q-k}\right] \tag{3.35}$$

$$= \frac{(1-f)\,\lambda^{k-1}}{(k-1)!}Z_B^{k-1}e^{-\lambda Z_B}\sum_{q=k}^{\infty}\frac{e^{-\lambda(1-Z_B)}\left[\lambda\left(1-Z_B\right)\right]^{q-k}}{(q-k)!} \tag{3.36}$$

$$-\frac{(1-f)\,\lambda^{k-1}}{(k-1)!}\left(Z_B-X_a\right)^{k-1}e^{-\lambda[1-(1-Z_B-X_b)]}\sum_{q=k}^{\infty}\frac{e^{-\lambda(1-Z_B-X_b)}\left[\lambda\left(1-Z_B-X_b\right)\right]^{q-k}}{(q-k)!}$$

$$\Leftrightarrow X_b = \frac{(1-f)\,\lambda^{k-1}}{(k-1)!}\left\{Z_B^{k-1}e^{-\lambda Z_B}-\left(Z_B-X_a\right)^{k-1}e^{-\lambda(Z_B+X_b)}\right\}. \tag{3.37}$$

Notice that X_a and X_b are coupled self-consistency equations, that we can solve using the algorithm presented in the section for the G-GCC, 2.3.3. We just have to be careful not to set both $X_a = X_b = 0$, since by Equation (3.37) we can see that this is always a solution, even though it is not always the stable one. However in order to always get the right solution one just can set the initial conditions as $X_a = X_b = \epsilon$, with ϵ being a small perturbation as $\epsilon = 0.0001$.

Having X_a and X_b we can calculate the fraction of the network that belongs to the BGCC, S_{X_B}. A way to construct S_{X_B} is to subtract from all active nodes, those that do not belong in the BGCC:

$$\begin{aligned} S_{X_B} =& S_{Z_B} - f\sum_{q=0}^{\infty}P(q)\sum_{l=0}^{q}\binom{q}{l}\left(Z_B-X_a\right)^{l}\left(1-Z_B-X_b\right)^{q-l} \\ &-(1-f)\sum_{q=k}^{\infty}P(q)\sum_{l=k}^{q}\binom{q}{l}\left(Z_B-X_a\right)^{l}\left(1-Z_B-X_b\right)^{q-l}. \end{aligned} \tag{3.38}$$

The first term corresponds to all the active nodes, S_{Z_B}.

The second term corresponds to the seeds, f, that do not connect to the BGCC in any way, i.e. seeds whose edges either lead to nodes that would be active independently of the existence of the seed, and that do not connect to the BGCC, $\left(Z_B-X_a\right)^{l}$; or to nodes that only became active because of the seed, but still do not connect to the BGCC $\left(1-\left(Z_B-X_b\right)^{q-l}\right)$. Notice that l starts in $l=0$, since seeds do not need a minimum number of active neighbours to become active.

The third term corresponds to the non-seed nodes $(1-f)$, that become active but that are not in the BGCC, i.e. nodes that have at leask k active neighbours that contributed to their activation, but who do not connect to the BGCC, $(Z_B - X_a)^l$; and out of the other edges, either they meet a not active node, or they meet a node which is active because of them, but that does not connect to the BGCC anyhow $(1 - Z_B - X_B)^{q-l}$.

We can simplify the equation for S_{X_B} in the following way:

$$\begin{aligned} S_{X_B} =& S_{Z_B} - f \sum_{q=0}^{\infty} P(q) \left(1 - X_B\right)^q \\ & - (1-f) \sum_{q=k}^{\infty} P(q) \sum_{l=k}^{q} \binom{q}{l} \left(Z_B - X_a\right)^l \left(1 - Z_B - X_b\right)^{q-l}. \end{aligned} \tag{3.39}$$

The term for the seeds is familiar to us, we have seen it for example in Equation (2.142). So we can substitute the result by performing the substitution $Y \to X_B$. The second term can be treated in a similar way as S_{Z_B}:

$$S_{X_B} = S_{Z_B} - f e^{X_B} - (1-f) \sum_{l=k}^{\infty} \sum_{q=k}^{\infty} P(q) \binom{q}{l} \left(Z_B - X_a\right)^l \left(1 - Z_B - X_b\right)^{q-l}, \tag{3.40}$$

which for an Erdős-Rényi graph reads:

$$S_{X_B} = \tag{3.41}$$

$$= S_{Z_B} - f e^{X_B} - (1-f) \sum_{l=k}^{\infty} \sum_{q=l}^{\infty} \frac{e^{-\lambda} \lambda^q}{q!} \frac{q!}{(q-l)!l!} \left(Z_B - X_a\right)^l \left(1 - Z_B - X_b\right)^{q-l}$$

$$= S_{Z_B} - f e^{X_B} - (1-f) \sum_{l=k}^{\infty} \frac{\left[\lambda \left(Z_B - X_a\right)\right]^l}{l!} e^{-\lambda(Z_B + X_b)} \sum_{q=l}^{\infty} \frac{e^{-\lambda(1 - Z_B - X_b)} \left[\lambda \left(1 - Z_B - X_b\right)\right]^{q-l}}{(q-l)!} \tag{3.42}$$

$$= S_{Z_B} - f e^{X_B} - (1-f) \sum_{l=k}^{\infty} \frac{\left[\lambda \left(Z_B - X_a\right)\right]^l}{l!} e^{-\lambda(Z_B + X_b)} \tag{3.43}$$

$$= S_{Z_B} - f e^{X_B} - (1-f) e^{-\lambda X_B} \sum_{l=k}^{\infty} \frac{e^{-\lambda(Z_B - X_a)} \left[\lambda \left(Z_B - X_a\right)\right]^l}{l!} \tag{3.44}$$

$$\Leftrightarrow S_{X_B} = S_{Z_B} - f e^{X_B} - (1-f) e^{-\lambda X_B} \left\{ 1 - \sum_{l=0}^{k-1} \left[\lambda \left(Z_B - X_a\right)\right]^l \frac{e^{-\lambda(Z_B - X_a)}}{l!} \right\} \tag{3.45}$$

$$\Leftrightarrow S_{X_B} = S_{Z_B} - e^{X_B} + (1-f) e^{-\lambda(Z_B + X_b)} \sum_{l=0}^{k-1} \frac{\left[\lambda \left(Z_B - X_a\right)\right]^l}{l!}. \tag{3.46}$$

Figure 3.4 shows the evolution of S_{X_B} with $\langle q \rangle$, for different values of k and f.

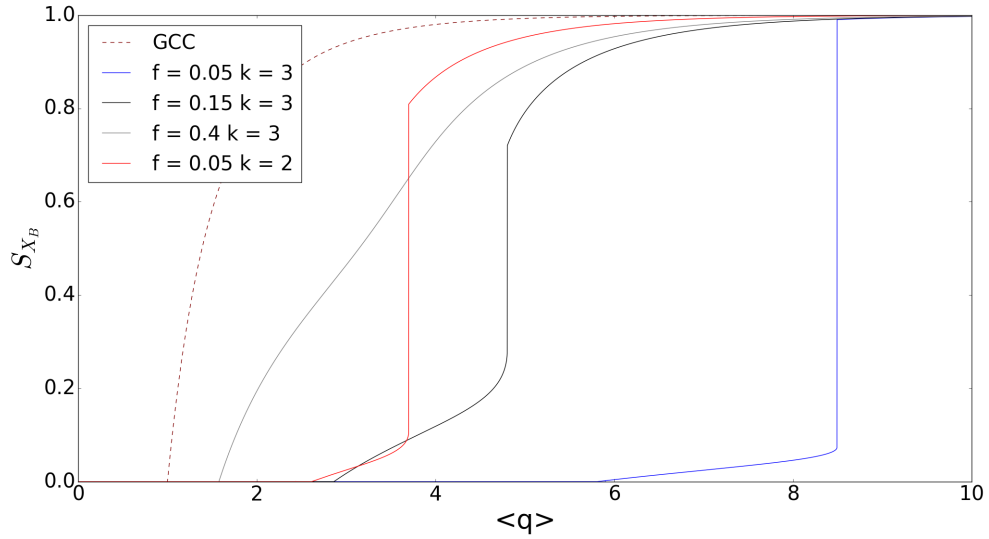


FIGURE 3.4: The evolution of S_{X_B} with $\langle q \rangle$, for different values of f and k. The dashed line represents the GCC, and the colored lines the size of the BGCC for the correspondent parameters.

The curves for S_{X_B} naturally increase with λ. This happens because the addition of edges increases the probability that more nodes join the BGCC, either by activation by members of the BGCC, or by connecting already active finite clusters to the BGCC.

As before, notice that the BGCC is smaller than the GCC. This is because the BGCC corresponds to the nodes that within the infinite GCC, make a fraction of it active. So the nodes that are added to the BGCC are always part of the GCC. In fact, the gray curve starts to have the shape of the curve for the GCC.

In the curves for S_{X_B} it is interesting that two types of phase transitions are observed. Like in the one for S_{Z_B}, there is a hybrid phase transition at λ_{crit2}, that corresponds to the activation of the subcritical clusters. Notice that the size of the jump in the curves for S_{X_B} and S_{Z_B}, has the same size. This is because the jump corresponds to an activation of an infinite number of nodes that belonged to the the subcritical clusters (remember that at λ_{crit2}, the mean subcritical cluster size diverges), since the size of the activated subcritical clusters is infinite, it is in the GCC.

Unlike the curve for S_{Z_B}, however, there is another phase transition that is observed at $\lambda_{crit1} < \lambda_{crit2}$, which corresponds to the appearance of the BGCC in the first place. λ_{crit1} is a critical point of a second order (continuous) phase transition, and happens when a second solution for X_B grows from zero. Physically, this point corresponds to the moment when the finite clusters of active nodes grow (with λ) and start to connect in a way that their sum corresponds to a finite fraction of the infinite network.

Once again, with the increase of f the curve moves to the left (or up), since, for the same $\langle q \rangle$, there are more seeds initially that activate more nodes in the GCC. Furthermore, there is a value when $f = f_{crit}$ where the hybrid transition ends.

With the increase of k, each non-seed node has a smaller probability of being active for the same $\langle q \rangle$, and f, making it seem that the curve moves to the right.

3.2.1 Limit $k = 1$

We have shown that we can generalize the equations for Z_B, X_a, X_b, X_B and S_{X_B}, for a percolation problem, by adding the condition to each of these equations that the edge was not removed (i.e. by multiplying by p). Let's define these new probabilities as Z_p, X_{p_a}, X_{p_b},X_p and S_{X_p}, respectively.

Notice that when $k = 1$, the percolation version of these equations should meet the ones for the SIR Model, since they are constructed in the same way, where p, the probability that we remove each edge, represents the probability of *usage* of an edge before recovering ($p = \frac{\beta(1-\alpha)}{\alpha+\beta(1-\alpha)}$). This mapping is consistent because in the SIR Model for a node to become infected (and later recovered) it just needs one neighbour ($k = 1$) to pass the infection to it, so the active nodes of the Bootstrap Percolation Model, in the infinite time limit, will correspond to being recovered in the SIR Model.

For the BGCC we point out that we should get the equations for the CRGCC and *not* for the G-GCC, due to the fact that in the Bootstrap Percolation Model a node can only be activated and connect to the BGCC via a not removed edge. In the Bootstrap Model, p represents the physical removal of the edge from the network, and not just the "not *usage*" of the edge as it does in the SIR Model, so an active node can never connect to the BGCC (nor to active nodes) via an edge that was removed, as it does in the G-GCC case. This also means that the Bootstrap Model does not have the conceptual problem of redundant edges, since a node that would be passed the infection twice (i.e. more that two infected nodes would use their edges to transmit the disease to it) in the SIR Model, is simply a node that connects to two active neighbours, that got active from different paths, in the Bootstrap Model.

The equations for Z_p and S_{Z_p} in this limit read:

$$Z_p = p\left[1 - (1-f)e^{-\lambda Z_p}\right] \tag{3.47}$$

$$S_{Z_p} = 1 - (1-f)e^{-\lambda Z_p}, \tag{3.48}$$

coinciding with the ones for the SIR Model.

For S_{X_B} in this limit we have:

$$S_{X_p} = S_{Z_p} - e^{-X_p} + (1-f)\, e^{-\lambda\left(Z_p + X_{p_b}\right)} \tag{3.49}$$

$$= 1 - (1-f)\, e^{-\lambda Z_p} - e^{-X_p} + (1-f)\, e^{-\lambda\left(Z_p + X_{p_b}\right)} \tag{3.50}$$

$$\Leftrightarrow S_{X_p} = 1 - e^{-X_p} - (1-f)\, e^{-\lambda Z_p}\left(1 - e^{-\lambda X_{p_b}}\right), \tag{3.51}$$

which meets the equation we derived for the CRGCC if $X_{p_b} = 0$.

However, what does X_{p_b} represent in this limit? In general X_{p_b} represents the probability that, being myself a node, that I follow one of my edges, and that I meet a node that got activated because of my activation, and that it connects to the BGCC, either via the other $k-1$ active nodes that contributed to its activation, or via a node that also got activated because of it. In this limit, $k - 1 = 0$. So the only possibility that I connect to the BGCC via a case of X_{p_b}, is if my neighbour also connects to the BGCC via X_{p_b}, and its neighbour as well,etc. This can only happen if there is only one seed in the whole BGCC, let's call it seed A. It means that the seed activated at least one of its neighbours, that activated another, and another, until a finite fraction of the Giant Connected Component became active. So it means that the entire BGCC was infected via a single seed, and the seed connects through the nodes it activated ($X_{p_b} \subset 1 - Z_p$), to the BGCC. Notice that if another seed, let's call it seed B, existed in the BGCC, all the active nodes would be active independently of seed A, so A would connect to the BGCC via a case of X_{p_a}.

In terms of the SIR model, this corresponds to seed A, starting an infection that spread until it formed a Giant Connected Component of Recovered nodes. Since all the nodes got infected from seed A, and seed A connects to the CRGCC (BGCC) via a node it infected. This enhances a second error committed in the construction of X_C. We had already seen that in the construction of this quantity there was some conceptual ambiguity regarding the condition that a seed could only connect to the CRGCC via a node that had *used* its edge in transmitting the disease to this seed. Now, in addition to that, we notice that just by taking these cases into account, we are missing the cases that the seed infected the whole CRGCC by itself, and connects to it via a $1 - Z$ case. Figure 3.5 presents one of these cases.

Notice that the cases of X_{p_b} are taken into account in the G-GCC, since $X_G \not\subset Z$. X_G is built by looking at the PRN in the infinite time limit, and accounts for all the nodes that connect to the G-GCC, whether they were transmitted the disease by it, whether they transmitted the disease to a node that then connects to it. In fact, term 3 of the cases of the X_G, Equation (2.105), and represented in Figure 2.20, includes nodes that connect to the G-GCC via other nodes that they transmitted the disease to.

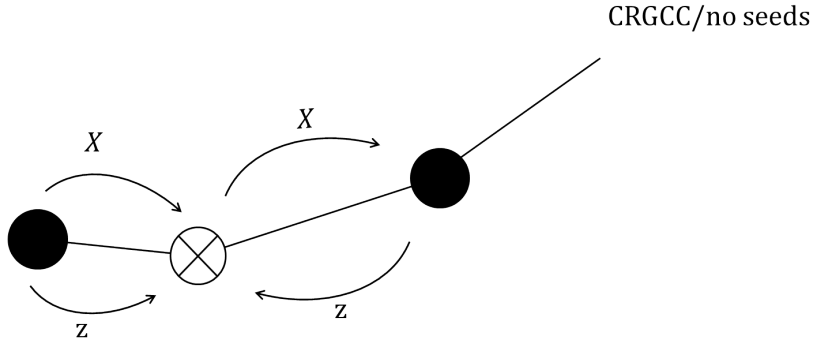


FIGURE 3.5: This figure shows a case where the seed infected the whole CRGCC, and connects to it via the nodes it infected, a case missed by the usual calculation of the CRGCC. The system for the meaning of the edges and nodes was the same as before, however, this time the black circles also mean an activated node (which has an equivalency for a recovered node).

However, it happens that the probability of these cases of X_{p_b} go to zero when $k = 1$ for an Erdős-Rényi graph. This explains why in the limit $f \to 0$, the G-GCC that accounts for these cases, meets the CRGCC that does not.

We present the proof that $X_{p_b} \to 0$, when $k = 1$:

$$X_{p_b} = (1-f)\sum_{q=1}^{\infty}\frac{qP(q)}{\langle q\rangle}\binom{q-1}{0}\left[Z_p^0\left(1-Z_B\right)^{q-1} - \left(Z_p - X_{p_a}\right)^0\left(1-Z_p-X_{p_b}\right)^{q-1}\right] \quad (3.52)$$

$$= (1-f)\sum_{q=1}^{\infty}\frac{qP(q)}{\langle q\rangle}\left[\left(1-Z_p\right)^{q-1} - \left(1-Z_p-X_{p_b}\right)^{q-1}\right], \quad (3.53)$$

where a first solution of $X_{p_b} = 0$ can be observed. Since we know the results of these summations from Equation (2.107), we can write X_{p_b} in this limit, in the form:

$$X_{p_b} = (1-f)e^{-\lambda Z_p}\left(1-e^{-\lambda X_{p_b}}\right) \equiv h\left(X_{p_b}\right) \quad (3.54)$$

Where $h\left(X_{p_b}\right)$ has been introduced has the right hand side of Equation (3.54). By looking at $h(X_{p_b})$, we can say that for a fixed λ it is a monotonically increasing function, bonded above by one, since the term $0 \leq (1-f)e^{-\lambda Z_p} \leq 1$ is constant, and the other $\left(1-e^{-\lambda X_{p_b}}\right)$ is monotonically increasing. This second term is the same term as for X_C in the CRGCC, Equation (2.92), whose derivative we know to be monotonically decreasing to zero, thus that the function is monotonically increasing to one.

Thus, we will have a second solution for X_{p_b} when $h\left(X_{p_b}\right)'|_{X_{p_b}=0} > 1$. If this condition is never met, it means that the only stable solution happens at $X_{p_b} = 0$. If we assume this we

get:

$$S_{X_p} = 1 - e^{X_p}, \tag{3.55}$$

where

$$X_{p_a} = p\left[1 - e^{\lambda X_p} - (1-f)e^{-\lambda Z_p}\left(1 - e^{-\lambda X_{p_b}}\right)\right] \tag{3.56}$$

$$\Rightarrow X_p = p\left(1 - e^{\lambda X_p}\right). \tag{3.57}$$

Which are the exact same equations for the CRGCC. Thus, the Bootstrap Model when $k = 1$ meets the CRGCC SIR Model, if $X_{p_b} = 0$ is the only stable solution. It will also be the only case of the Bootstrap Model that the BGCC does not depend on the seed.

3.3 The Bootstrap Model's Critical Points

3.3.1 Calculation of λ_{crit1}

We will now present a way to calculate the critical points for the Bootstrap Model.

In order to calculate λ_{crit1}, one can use a similar approach in previous chapters, noticing that both the solution for X_B, appears from zero. When this solution starts to appear we have that $X_B \ll 1 \Leftrightarrow X_a, X_b \ll 1$, since $X_a, X_a \in [0, 1]$.

In this work we will calculate these critical points for $k = 2$ and $k = 3$, since these were the values we used in our simulations, although any other value of k could be calculated using a similar approach.

We can obtain the value for $k = 1$, from our mapping to the SIR:

$$\lambda_{crit1} = \frac{1}{p} = 1 \tag{3.58}$$

Meaning the whole GCC will get infected. This comes as a natural result since by the existence of a single seed, all the nodes that belong to its cluster will become infected.

For $k = 3$, if we make a Taylor expansion around $X_B = 0$, we get:

$$X_a \simeq Z_B - \left[1 - \lambda X_B + O\left(X^2\right)\right] + (1-f)\, e^{-\lambda(Z_B)} \left[1 - \lambda X_b + O\left(X_b\right)\right] \left\{1 + \lambda\left(Z_B - X_a\right) \frac{\left[\lambda\left(Z_B - X_a\right)\right]^2}{2}\right\}. \tag{3.59}$$

Disregarding 2nd order terms:

$$X_a = Z_B - (1 - \lambda X_B) + (1-f)\, e^{-\lambda(Z_B)} (1 - \lambda X_b) \left[1 + \lambda\left(Z_B - X_a\right) \frac{\lambda^2 Z_B^2}{2} - \frac{2\lambda^2 Z_B X_a}{2}\right] \tag{3.60}$$

$$= Z_B - (1 - \lambda X_B) + (1-f)(1 - \lambda X_b) \left[-\lambda\left(X_a\right) e^{-\lambda(Z_B)} - \lambda^2 Z_B X_a e^{-\lambda(Z_B)} - \frac{Z_B - 1}{(1-f)}\right], \tag{3.61}$$

where the substitution $Z_{Boot_{k=3}} = 1 - (1-f)\, e^{-\lambda Z_{Boot_{k=3}}} \left(1 + \lambda Z_{Boot_{k=3}} + \frac{\lambda Z_{Boot_{k=3}}^2}{2!}\right)$ was made. Then

$$X_a = Z_B - 1 + \lambda X_B + 1 - \lambda X_b - Z_B + \lambda X_b Z_B - (1-f)\, \lambda X_a e^{-\lambda Z_B} (1 + \lambda Z_B) \tag{3.62}$$

$$= \lambda X_a + \lambda X_b - (1-f)\, \lambda X_a e^{-\lambda Z_B} (1 + \lambda Z_B) \tag{3.63}$$

$$\Leftrightarrow X_b = \frac{X_a}{\lambda Z_B}\left[1 - \lambda + (1-f)\, \lambda e^{-\lambda Z_B} (1 + \lambda Z_B)\right]. \tag{3.64}$$

For X_b we have:

$$X_b = \frac{(1-f)\,\lambda^{k-1}}{(k-1)!}\left\{Z_B^{k-1} e^{-\lambda Z_B} - \left(Z_B - X_a\right)^{k-1} e^{-\lambda(Z_B + X_b)}\right\} \tag{3.65}$$

$$= \frac{(1-f)\,\lambda^{k-1}}{(k-1)!}\left\{Z_B^{k-1} e^{-\lambda Z_B} - \left(Z_B - X_a\right)^{k-1} e^{-\lambda(Z_B)} \left[1 - \lambda X_b + O\left(X_b^2\right)\right]\right\} \tag{3.66}$$

$$\simeq \frac{(1-f)\,\lambda^{k-1}}{(k-1)!} e^{-\lambda Z_B}\left\{Z_B^{k-1} - \left(Z_B - X_a\right)^{k-1} (1 - \lambda X_b)\right\} \tag{3.67}$$

$$= \frac{(1-f)\,\lambda^2}{(2)!} e^{-\lambda Z_B}\left\{Z_B^2 - \left(Z_B^2 - Z_B^2 \lambda X_b - 2 X_a Z_B\right)(1 - \lambda X_b)\right\} \tag{3.68}$$

$$\Leftrightarrow X_b = \frac{(1-f)\, Z_B \lambda^2 e^{-\lambda Z_B}}{2}\left\{Z_B \lambda X_b + 2 X_a\right\}. \tag{3.69}$$

From which we can write:

$$X_a = X_b \left(\frac{1}{(1-f)\, Z_B \lambda^2 e^{-\lambda Z_B}} - \frac{\lambda Z_B}{2} \right). \tag{3.70}$$

Putting together both these conditions we get:

$$0 = \frac{1}{\lambda Z_B} \left[1 - \lambda + (1-f)\, \lambda e^{-\lambda Z_B} (1 + \lambda Z_B) \right] \left(\frac{1}{(1-f)\, Z_B \lambda^2 e^{-\lambda Z_B}} - \frac{\lambda Z_B}{2} \right) - 1. \tag{3.71}$$

Notice that this equation is solely a function of λ since we can determine Z_B in a self-consistent way given λ. Thus, we can determine its solution using a bisection method like the one introduced in section 2.3.5. We only need to be careful to re-calculate Z every time we use a new value of λ. Since the function is strictly positive for $\lambda < \lambda_{crit1}$, and strictly negative for $\lambda > \lambda_{crit1}$, as we can see in Figure 3.6, we can set the initial conditions as $a = 0$, and $b = 100$, since in the range we explored, we never got to $\langle q \rangle = 100$.

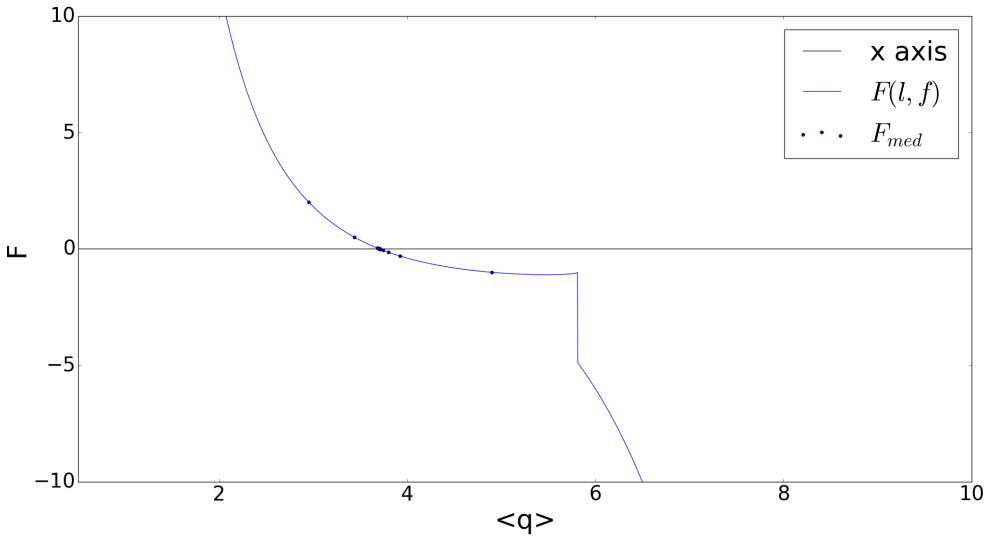


FIGURE 3.6: This figure shows the evolution of the right hand side of Equation (3.71), represented by F, the blue line. The blue dots represent the successive values of F_{med}, the value of F that the algorithm would calculate each step.

Applying the same procedure to $k = 2$ we get:

$$X_a = Z_B - e^{-\lambda X_B} + (1-f)\, e^{-\lambda(Z_B+X_b)} \left[1 + \lambda\left(Z_B - X_a\right)\right] \tag{3.72}$$

$$= Z_B - \left[1 - \lambda X_B + O(X_B^2)\right] + (1-f)\, e^{-\lambda Z_B} \left[1 - \lambda X_b + O(X_b^2)\right] \left[1 + \lambda\left(Z_B - X_a\right)\right] \tag{3.73}$$

$$\simeq Z_B - 1 + \lambda X_B + (1-f)\, e^{-\lambda Z_B} \left(1 - \lambda X_b\right)\left(1 + \lambda Z_B - \lambda X_a\right) \tag{3.74}$$

$$= Z_B - 1 + \lambda X_B + (1-f)\, e^{-\lambda Z_B} \left[1 - \lambda X_b + \lambda Z_B - \lambda^2 Z_B X_b - \lambda X_a + O\left(X_b X_a\right)\right] \tag{3.75}$$

$$= Z_B - 1 + \lambda X_B + (1-f)\, e^{-\lambda Z_B} \left(1 + \lambda Z_B\right) \tag{3.76}$$

$$- \lambda X_b \left(1-f\right) e^{-\lambda Z_B} \left(1 + \lambda Z_B\right) - \lambda X_a \left(1-f\right) e^{-\lambda Z_B}$$

$$= Z_B - 1 + \lambda X_B + 1 - Z_B - \lambda X_b \left(1 - Z_B\right) - \lambda X_a \left(1-f\right) e^{-\lambda Z_B} \tag{3.77}$$

$$= Z_B - 1 + \lambda X_B + 1 - Z_B - \lambda X_b \left(1 - Z_B\right) - \lambda X_a \left(1-f\right) e^{-\lambda Z_B} \tag{3.78}$$

$$= \lambda X_a \left[1 - \left(1-f\right) e^{-\lambda Z_B}\right] + \lambda X_b Z_B \tag{3.79}$$

$$\Leftrightarrow X_b = \frac{X_a}{\lambda Z_B} \left\{1 - \lambda \left[1 - \left(1-f\right) e^{-\lambda Z_B}\right]\right\}. \tag{3.80}$$

Then for X_b:

$$X_b = \lambda\left(1-f\right) \left[Z_B e^{-\lambda Z_B} - \left(Z_B - X_a\right) e^{-\lambda\left(Z_B - X_b\right)}\right] \tag{3.81}$$

$$= \lambda\left(1-f\right) e^{-\lambda Z_B} \left[Z_B - \left(Z_B - X_a\right) e^{-\lambda X_b}\right] \tag{3.82}$$

$$= \lambda\left(1-f\right) e^{-\lambda Z_B} \left[Z_B - \left(Z_B - X_a\right)\left(1 - \lambda X_b + O\left(X_b^2\right)\right)\right] \tag{3.83}$$

$$\simeq \lambda\left(1-f\right) e^{-\lambda Z_B} \left[Z_B - Z_B + X_a - \lambda Z_B X_b + O\left(X_b X_a\right)\right] \tag{3.84}$$

$$\simeq \lambda\left(1-f\right) e^{-\lambda Z_B} \left[X_a - \lambda Z_B X_b\right] \tag{3.85}$$

$$\Leftrightarrow X_a = X_b \left[\lambda Z_B + \frac{1}{\lambda\left(1-f\right) e^{-\lambda Z_B}}\right]. \tag{3.86}$$

Combining Equations (3.80) and (3.86), we reach the condition for λ_{crit1}:

$$0 = \frac{1}{\lambda Z_B} \left\{1 - \lambda \left[1 - \left(1-f\right) e^{-\lambda Z_B}\right]\right\} \left[\lambda Z_B + \frac{1}{\lambda\left(1-f\right) e^{-\lambda Z_B}}\right] - 1 \equiv F(\lambda) \tag{3.87}$$

whose zero we can determine using our bisection method. Once again, we can set the initial conditions as $a = 0$, and $b = 100$, since in the range we explored numerically, we never got needed to get to $\langle q \rangle = 100$, let alone the critical point.

3.3.2 Calculation of λ_{crit2}

For the calculation of the hybrid critical point, λ_{crit2}, one has to determine the appearance of the second solution of Z_B. Defining $\Psi(Z)$ as the right hand side of Equation (3.16), the second solutions appears when the following conditions are met:

$$\Psi(Z) = Z_B \tag{3.88}$$

$$\Psi'(Z) = Z'_B = 1 \Leftrightarrow \left(\frac{\Psi(Z)}{Z_B}\right)' = 0. \tag{3.89}$$

Notice that there are two values of λ in which both conditions are fulfilled, one when the second solution first appears, and another when it disappears. Figure 3.7 shows these two cases.

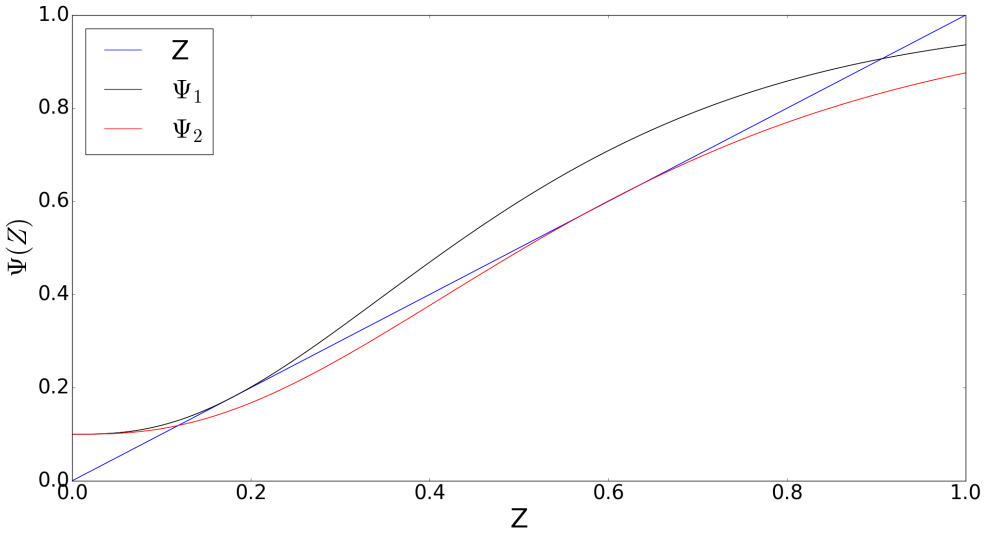


FIGURE 3.7: This figure shows the evolution of $\Psi(Z)$ with Z. The black and red line cross Z at the points that satisfy conditions (3.88) and (3.89). The black line crosses Z at λ_{crit2}, and the red line at the equivalent of λ_{crit2} if we were analyzing a system of k-Core Percolation.

However, when we solve these self consistency equations numerically, we will obtain the right solution, the smallest of them both, since we are calculating Z_B always starting at $LHS = 0$, for any λ.

Calculating $\left(\frac{\Psi(Z)}{Z_B}\right)'$ for $k = 2$:

$$\Psi(Z) = 1 - (1-f)\,(1+\lambda Z_B)\,e^{-\lambda Z_B} \tag{3.90}$$

$$\Leftrightarrow \left(\frac{\Psi(Z)}{Z}\right)' = -\frac{1}{Z_B^2} - (1-f)\,e^{-\lambda Z_B}\left[-\frac{1}{Z_B^2} - \lambda\left(\frac{1}{Z_B} + \lambda\right)\right] \tag{3.91}$$

$$= -\frac{1}{Z_B^2} + \frac{1}{Z_B^2}\,(1-f)\,e^{-\lambda Z_B} + \frac{\lambda}{Z_B}\,(1+\lambda Z_B)\,(1-f)\,e^{-\lambda Z_B} \tag{3.92}$$

$$= -\frac{1}{Z_B^2} + \frac{1}{Z_B^2}\,(1-f)\,e^{-\lambda Z_B} + \frac{\lambda}{Z_B}\,(1+\lambda Z_B)\,(1-f)\,e^{-\lambda Z_B} \tag{3.93}$$

$$= -\frac{1}{Z_B^2}\left[1 - (1-f)\,e^{-\lambda Z_B}\right] + \frac{\lambda}{Z_B}\lambda\,(1-Z_B) \tag{3.94}$$

$$= -\frac{1}{Z_B^2}\left[Z_B + \lambda Z_B\,(1-f)\,e^{-\lambda Z_B}\right] + \frac{\lambda}{Z_B}\,(1-Z_B) \tag{3.95}$$

$$= -\frac{1}{Z_B}\left[1 + \lambda\,(1-f)\,e^{-\lambda Z_B} - \lambda\,(1-Z_B)\right]. \tag{3.96}$$

Since $\left(\frac{\Psi(Z)}{Z}\right)' = 0$ at λ_{crit2}:

$$\Rightarrow 0 = 1 + \lambda\,(1-f)\,e^{-\lambda Z_B} - \lambda\,(1-Z_B) = \Phi(Z). \tag{3.97}$$

So, in order to calculate λ_{crit2} we just have to calculate the value of λ that satisfies conditions (3.97). However this time we cannot use a bisection method since the function on the right hand side of Equation (3.97), Φ, is positive on both sides of λ_{crit2}, and only meets zero at exactly $\lambda = \lambda_{crit2}$. We plot a figure of that shows how Φ evolves with λ:

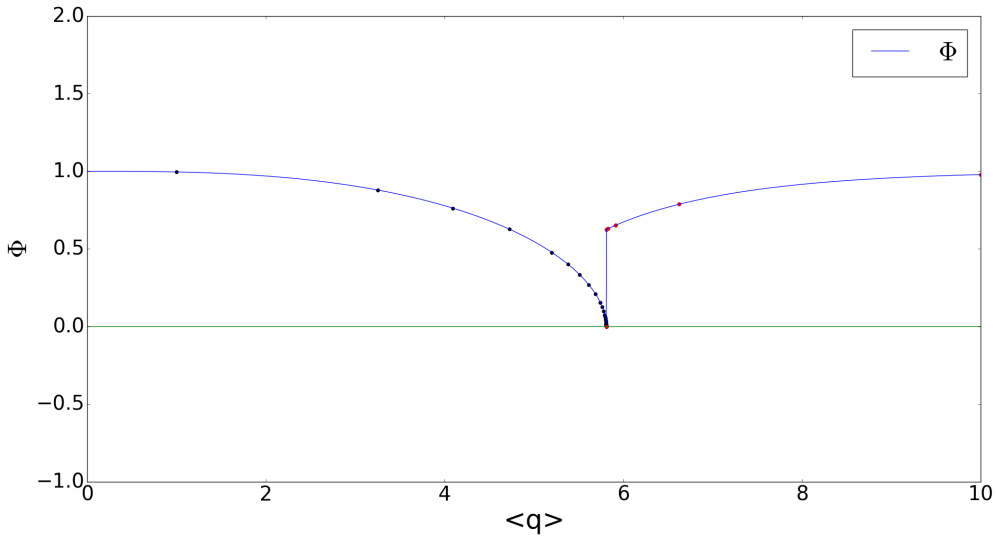


FIGURE 3.8: This figure shows the evolution of Φ with $\langle q \rangle$, for $k = 3$ and $f = 0.1$. The black and red dots represent the successive values of Φ_{min} and Φ_{max}, respectively.

In order to solve this we created an algorithm that would find minima of functions. The algorithm we constructed only works in an interval around λ_{crit2}, with only one local minimum, which in our case happened for every value of λ.

The algorithm worked in the following way: Set two values λ_{min} and λ_{max}, which would set the interval in which we are looking for the minimum. Since we know that $\lambda_{crit2} > 1$, we set $\lambda_{min} = 1$, and $\lambda_{max} = 100$, to ensure we find a solution. We calculate the value of Φ in these points obtaining Φ_{min} and Φ_{max}, respectively (in this algorithm everytime a new value of λ was used, we would have to re-calculate Z_B as well). Then, we would calculate the middle point $\lambda_{med} = \frac{\lambda_{min}+\lambda_{max}}{2}$, and Φ_{med}, the value of RHS in this point. Finally, since the function on the right side is monotonically decreasing we would adapt $\lambda_{max} = \lambda_{med}$, everytime that $\Phi_{med} < \Phi_{max}$. This would make us always approximate the right boundary of our interval to λ_{crit2}, with one exception: that λ_{med} was on the left side of λ_{crit2} and with $\Phi_{med} < \Phi_{max}$. In this case however, in the following iteration, Φ_{med} would necessarily be bigger than Φ_{max}, since we would be walking towards the left boundary of the interval. So, in order to account for this case, everytime that $\Phi_{med} \geq \Phi_{max}$ we would attribute $\lambda_{min} = \lambda_{med}$ (since we know λ_{med} would be on the left side of λ_{crit2}), and make λ_{max} retake the value it had in the previous iteration, i.e. the iteration before it was adapted to λ_{med} that is on the left side of λ_{crit2} (we should warn that depending on how we are keeping track of the values of λ_{max} we might need to update not only the one from this iteration, but also the one from the previous one, in order to make sure that all the values of λ_{max} that we are keeping are on the right hand side of λ_{crit2}. Otherwise, if we just update the value for this iteration, if again λ_{med} is on the left side of λ_{crit}, we might be re-adapting λ_{max} to the value that is on the left hand side).

Using the same procedure for $k = 3$, we get:

$$\left(\frac{\Psi(Z)}{Z_B}\right) = \frac{1}{Z_B} - (1-f)\,e^{-\lambda Z_B}\left[\frac{1}{Z_B} + \lambda + \frac{\lambda^2 Z_B}{2}\right] \tag{3.98}$$

$$\Leftrightarrow \left(\frac{\Psi(Z)}{Z_B}\right)' = -\frac{1}{Z_B^2} - (1-f)\,e^{-\lambda Z_B}\left[-\lambda\left(\frac{1}{Z_B} + \lambda + \frac{\lambda^2 Z_B}{2}\right) + \left(-\frac{1}{Z_B^2}\right) + \frac{\lambda^2}{2}\right] \tag{3.99}$$

$$= -\frac{1}{Z_B^2} - (1-f)\,e^{-\lambda Z_B}\left[\frac{\lambda^2}{2} + \frac{\lambda}{Z_B} + \frac{\lambda^3 Z_B^3}{2} + \frac{1}{Z_B^2}\right] \tag{3.100}$$

$$\Leftrightarrow 0 = 1 - (1-f)\,e^{-\lambda Z_B}\left[1 + \lambda Z_B + \left(\frac{\lambda Z_B}{2}\right)^2\right] - (1-f)\,e^{-\lambda Z_B}\frac{\lambda^3 Z_B}{2}. \tag{3.101}$$

Noticing that $Z_B = 1 - (1-f)\,e^{-\lambda Z_B}\left[1 + \lambda Z_B + \left(\frac{\lambda Z_B}{2}\right)^2\right]$, we obtain the condition:

$$\Leftrightarrow 0 = 1 - (1-f)\, e^{-\lambda Z_B} \frac{\lambda^3 Z_B^2}{2} \tag{3.102}$$

Where we can solve using the same minima finding algorithm as for $k = 2$.

3.3.3 End of λ_{crit2}

As we could see in the curves for S_{X_B}, the hybrid transition does not always exist. In order to determine the value of f where it disappears, f_{crit}, we have to introduce a third condition in Equations (3.97) and (3.102). This can determined by looking at the solutions for Z_B with increasing f, and noticing when f increases, in the regime where there are two solutions, the difference between the two diminishes. At f_{crit}, the difference is zero, and the solution for Z_B evolves continuously. This implies that f_{crit} is the point when the two solutions for λ_{crit2} join, meaning it is an inflection point: $\Psi(Z)'' = 0$. Figure 3.9represents this case.

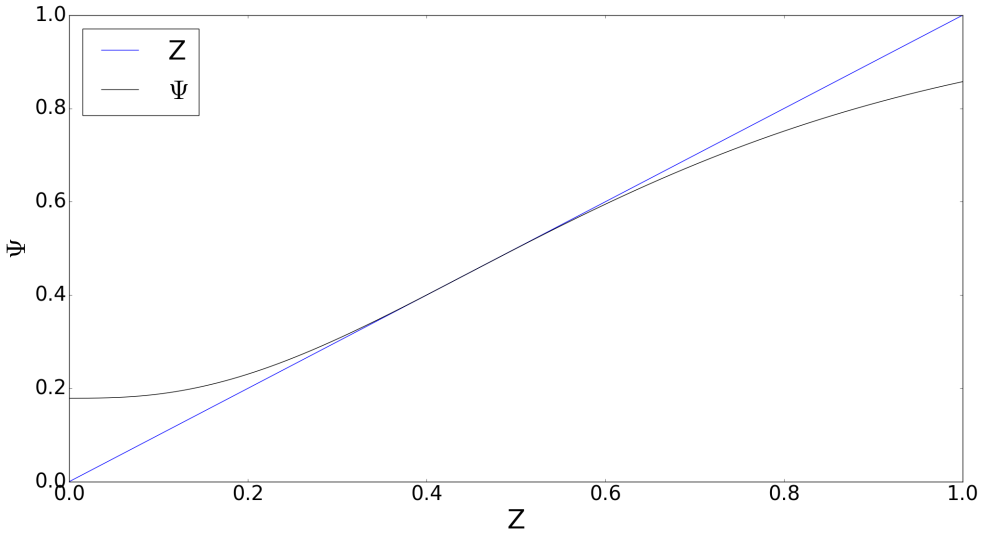


FIGURE 3.9: This figure shows the evolution of Ψ with Z, for $k = 3$, at the point where the two solutions for Equation (3.102) meet.

Thus we reach three conditions:

$$Z_{Boot} = \Psi(Z_B) \tag{3.103}$$

$$\left(\frac{\Psi(Z_B)}{Z_B}\right)' = 0 \tag{3.104}$$

$$\Psi(Z_B)'' = 0. \tag{3.105}$$

Let's start by calculating the case of $k = 3$, Equation (3.103) becomes

$$Z_B = 1 - (1-f)e^{-\lambda Z_B}\sum_{q=0}^{k-1}\frac{(\lambda Z_B)^q}{q!} \equiv \Psi(Z_B), \tag{3.106}$$

where we have three variables (λ, Z_B, and f) and three equations to solve for them.

From the condition (3.104) we have already obtained that

$$0 = 1 - (1-f)e^{-\lambda Z_B}\frac{\lambda^3 Z_B^2}{2}. \tag{3.107}$$

From condition (3.105) we get:

$$\Psi(Z_B)'' = -(1-f)e^{-\lambda Z_B}(-\lambda)\left\{-\lambda\left[1+\lambda Z_B+\frac{(\lambda Z_B)^2}{2}\right]+(\lambda+\lambda^2 Z_B)\right\} \tag{3.108}$$

$$-(1-f)e^{-\lambda Z_B}[-\lambda(\lambda+\lambda^2 Z_B)+\lambda^2]$$

$$= -(1-f)e^{-\lambda Z_B}\left(\lambda^2+\lambda^3 Z_B+\frac{\lambda^4 Z_B^2}{2}-\lambda^2-\lambda^3 Z_B+-\lambda^2-\lambda^3 Z_B+\lambda^2\right) \tag{3.109}$$

$$= -(1-f)e^{-\lambda Z_B}\left(\frac{\lambda^4 Z_B^2}{2}-\lambda^3 Z_B\right) \tag{3.110}$$

$$\Psi(Z_B)'' = -(1-f)e^{-\lambda Z_B}\lambda^3 Z_B\left(\frac{\lambda Z_B}{2}-1\right), \tag{3.111}$$

which at f_{crit} reads:

$$-(1-f)e^{-\lambda Z_B}\lambda^3 Z_B\left(\frac{\lambda Z_B}{2}-1\right) = 0 \tag{3.112}$$

$$\Rightarrow \lambda Z_B = 0 \vee \frac{\lambda Z_B}{2}-1 = 0 \tag{3.113}$$

$$\Leftrightarrow Z_B = \frac{2}{\lambda}. \tag{3.114}$$

So, we get a relation between λ and Z_B that satisfies condition (3.105). Plugging this result into the second condition, Equation (3.107):

$$0 = 1 - (1-f)e^{-\lambda\frac{2}{\lambda}}\frac{\lambda^3}{2}\frac{2^2}{\lambda^2} = 1 - (1-f)e^{-2}2\lambda \tag{3.115}$$

$$\Leftrightarrow 2(1-f)e^{-2}\lambda = 1 \tag{3.116}$$

$$\Leftrightarrow \lambda = \frac{1}{2(1-f)e^{-2}}. \tag{3.117}$$

We thus obtain a relation between λ and f that satisfies both condition (3.107) and (3.105). Substituting the result from equation (3.114) in (3.106):

$$Z_B = \frac{2}{\lambda} = 1 - (1-f)e^{-2}(1+2+2) = 1 - 5(1-f)e^{-2} \tag{3.118}$$

$$\Leftrightarrow \lambda = \frac{2}{1-5(1-f)e^{-2}}, \tag{3.119}$$

and combining the results from matching conditions (3.114) and (3.117), we obtain:

$$\frac{1}{2(1-f)e^{-2}} = \frac{2}{1-5(1-f)e^{-2}} \tag{3.120}$$

$$\Leftrightarrow 1 - 5(1-f)e^{-2} = 4(1-f)e^{-2} \tag{3.121}$$

$$\Leftrightarrow 1 = 9(1-f)e^{-2} \tag{3.122}$$

$$\Leftrightarrow f = 1 - \frac{e^2}{9} = 0.17899... \equiv f_{crit_{k=3}}. \tag{3.123}$$

Hence we obtain the value for f that determines when λ_{crit2} disappears. $f_{crit_{k=3}}$ is the point where there is no longer a jump; for any $f < f_{crit_{k=3}}$, there is a jump.

For $k = 2$. we have:

$$\Psi(Z) = 1 - (1-f)\, e^{-\lambda Z_B}\,(1+\lambda Z_B) \tag{3.124}$$

$$\Psi(Z)' = -\,(1-f)\, e^{-\lambda Z_B}\,[-\lambda\,(1+\lambda Z_B) + \lambda] \tag{3.125}$$

$$\Psi(Z)'' = \lambda^2\,(1-f)\, e^{-\lambda Z_B} + Z_B\lambda^2\,(1-f)\, e^{-\lambda Z_B}\,(-\lambda) \tag{3.126}$$

$$\Psi(Z)'' = \lambda^2\,(1-f)\, e^{-\lambda Z_B}\,(1 - Z_B\lambda)\,, \tag{3.127}$$

which at f_{crit} reads:

$$0 = \lambda^2 (1-f) e^{-\lambda Z_B} [1 - Z_B \lambda] \tag{3.128}$$

$$\Leftrightarrow \lambda = 0 \vee 1 - Z_B \lambda = 0 \tag{3.129}$$

$$\Leftrightarrow Z_B = \frac{1}{\lambda}. \tag{3.130}$$

Combining this condition with Equation (3.102):

$$0 = 1 - \lambda^2 \frac{1}{\lambda} (1-f) e^{-\lambda \frac{1}{\lambda}} = 1 - \lambda (1-f) e^{-1} \tag{3.131}$$

$$\Leftrightarrow \lambda = \frac{e}{(1-f)}, \tag{3.132}$$

which substituting in the equation for Z_B

$$\frac{1}{\lambda} = 1 - (1-f) \left(1 + \lambda \frac{1}{\lambda}\right) e^{-\lambda \frac{1}{\lambda}} = 1 - 2(1-f) e^{-1}. \tag{3.133}$$

Using Equations (3.132) and (3.133):

$$\frac{e}{(1-f)} = 1 - 2(1-f) e^{-1} \tag{3.134}$$

$$\Leftrightarrow e \left[1 - 2(1-f) e^{-1}\right] = (1-f) \tag{3.135}$$

$$\Leftrightarrow e = 3(1-f) \tag{3.136}$$

$$\Leftrightarrow f = 1 - \frac{e}{3} \equiv f_{crit_{k=2}} \approx 0.0939. \tag{3.137}$$

Notice that $f_{crit_{k=2}} < f_{crit_{k=3}}$.

Figure 3.10 shows the curves for the evolution of λ_{crit1} and λ_{crit2}, for both $k = 3$ and $k = 2$, with f. The lines represent the value of $\langle q \rangle$ when each transition happened.

The curves represented before for S_{X_B} with $\langle q \rangle$, for a fixed f, such as the curves from Figure 3.4, correspond in Figure 3.10 to following a vertical line at a fixed f. As we cross the plotted lines each transition happens.

Notice that the black and blue lines, representing λ_{crit1}, decrease as we increase f. This is due, as we explored in the curves of S_{X_B}, to the fact that as there are more seeds, each node will have a higher chance of meeting k active neighbours. Thus, the active components will be bigger for bigger values of f, in graphs with the same $\langle q \rangle$. As the active components are

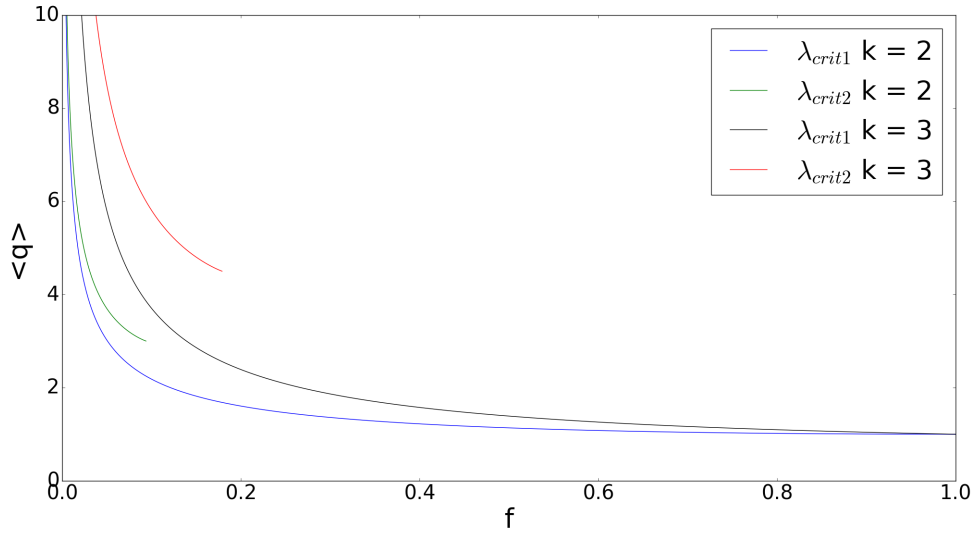


FIGURE 3.10: This figure shows the evolution of the critical points with $\langle q \rangle$, for $k = 2$ and $k = 3$.

bigger, the moment in which they connect to each other forming an infinite size component, happens sooner (going back to our analogy, the moment in which the fires meet).

The green and red lines, representing λ_{crit2}, also decreases as we increase f for a similar reason. Since each node has a higher probability of meeting active nodes, the subcritical clusters will diverge for smaller values of $\langle q \rangle$, i.e. what would require a value $\langle q \rangle_1$ to form a subcritical cluster, for f_1, happens at $\langle q \rangle_2 < \langle q \rangle_1$, for $f_2 > f_1$. The addition of edges is in a way compensated for the addition of seeds. The previously discussed f_{crit} corresponds to the end of the black line.

Finally, notice that both the transitions happen earlier for smaller k, due to the fact that each node becomes more easily active (naturally if a node has three active neighbours, it necessarily as two as well; the reverse is not true, so all the nodes that are active in a network where $k = 3$, would be active if $k = 2$). As such in order to obtain the same behavior of the network, for bigger k, either $\langle q \rangle$ or f need to be bigger. Making the critical phenomena also appear for higher values of these quantities.

Chapter 4

Numerical Methods

In this study, we did not manage to find an analytic way to describe the SIR-Bootstrap Model, other than in the limits where it converges to the known SIR and Bootstrap Models. The difficulty appeared in the time dependence of the equations from one iteration to the other. In this model it is not possible to look at the network in the infinite time limit, and define a quantity such as p as in the standard SIR case (see section 2.2). We cannot do this, since the infection of each node is not just given by the probability that each infected neighbour *used* an edge independently; the infected neighbours have to make successful contacts at the same time. Several attempts were made into creating time dependent equations, but ultimately a general analytic description was not found.

However, through numerical simulations it was possible to understand the general behaviour of the system qualitatively, such as the appearance and evolution of critical points, with different values of α, β, f, and k. In fact, we developed numerical methods that measured when and if each curve had a critical point. Naturally, all our numerical results are subject to errors, nevertheless, we believe them to be close to the "real theory", based on their agreement with the theoretical predictions in the known limits (SIR and Bootstrap model).

4.1 Simulation of the SIR-Bootstrap Model

The simulation of the SIR-Bootstrap Model was developed using the Python programming language [36] and the Jupyter Notebook [37]. In particular, the numpy [38], networkx [39], matplotlib [40], scipy libraries [41] were used, as well as the module os.

At the start the following quantities would be defined: N, the number of nodes of the network we would consider; λ the mean degree of the Erdős-Rényi graph we would generate; α, the probability of recovery of an infected node each iteration; β the probability that a contact between a susceptible node and an infected one is successful; k, the number of susceptible

contacts needed each time, for a susceptible node to jump to the infected state; f, the fraction of seeds in the network.

Then, an Erdős-Rényi graph, $G(N, r)$, would be generated, using the networkx's function: *fast_gnp_random_graph()*. Where r is defined as $r = \frac{\lambda}{N}$, as we have seen in the section for Random Networks , section 2.1.

Afterwards, a copy of $G(N, r)$ would be made, so to keep the original structure of the Graph, we defined this copy as *Recovered_Final()*. Also, three new, empty, graphs would be generated: $I.()$, a graph that would keep track of the nodes in the Infected state each iteration; $I.temp()$, a graph that would keep track of the change in the infected nodes, within the same iteration (we will explain in more detail the necessity of the introduction of this graph later on); and $R()$ a graph that would keep track of the recovered nodes each iteration.

After this, we selected the nodes that would be seeds. In order to do this, we calculated the quantity $N_{i0} = int(f \times N)$, the number nodes that would be selected to be seeds (notice that N_{i0} is an integer). Later, we would create an array with the identity of all the nodes, uniformly choose one, select it as a seed, and remove it from the array. We would do this N_{i0} times, obtaining a uniform randomly selected fraction of the network, the seeds. When we refer to "select it as a seed", we mean to introduce this node in the $I()$ and $I.temp()$ graphs, giving these nodes the state of Infected.

Another way to select the seeds would be to attribute a probability f to each node of being selected as a seed, and then checking those who were. This way N_{i0} would be a random variable, following a Binomial distribution, that through the law of Large numbers should tend to $f \times N$, when N is very big. We decided that fixing N_{i0} would minimize the system's numeric fluctuations. Furthermore, the only way $\frac{N_{i0}}{N}$ is different than the desired f, is when the product $f \times N$ is not an integer. In this case, we would round down N_{i0}, making at most N_{i0} miss one node to attribute as a seed, which is negligible when $N \gg 1$, and is actually the minimal possible error obtainable.

After the system is set, we would introduce the time variable, through a while loop, that would repeat the following procedure until there were no more infected nodes ($I.nodes() = []$). The system would evolve using synchronous update (all the changes in an iteration would be updated in the beginning of the next one, making all nodes evolve "at the same time"). We should note that we decided to first check the nodes which had recovered, and only then those which had been infected. This choice is just as valid as its opposite (first checking those that got infected, and only then those which had recovered), since the whole behaviour of the system is reachable in both cases by changing α and β. An example of this is the limit $\alpha = 1$ and $\beta = 1$ (an infinite infection and recovery rate), where with our definition, all the seeds will immediately recover, and with the opposite definition, all the nodes would

get infected. However, we can obtain the limit in which all the nodes end infected (and then recovered), using our definition, by decreasing α (a similar argument can be applied for the limits of the opposite definition).

Firstly, we would go through the nodes that existed in $I()$, using a for loop. For each, we would generate a sample, r_2, selected from a uniform random distribution in the interval $[0, 1[$, using numpy's *np.random.random_sample()* function. Then, we would compare r_2 with α, considering, `if r_{2} < α`, that the node had recovered (Notice that in the way r_2 is generated, the probability of each infected node to recover is α). When a node recovered, it would be removed from $I()$, $I.temp()$ and $G(N, r)$, and introduced into $R()$. The removal of each recovered node from $G(N, r)$, was chosen because a recovered node has no dynamics, it cannot be infected anymore, and does not interact with its neighbours. So, the dynamics on the network, having a node recovered, evolve in the same way as if that node did not exist in the network in the first place. This way, $G(N, r)$, can be seen as the dynamics graph, that keeps track only of nodes who can interact with each other, i.e. susceptible nodes and infected nodes. Furthermore, the removal of the recovered nodes from $G(N, r)$, speeds up our simulations, as we will see. Note that once a node is removed, all its edges are removed as well.

Later, in order to simulate the susceptible nodes that would become infected, we noticed that a susceptible node can only be infected if it has at least one infected neighbour. So, we would choose each infected node at a time, $infec$ (a node of $I()$), using a for loop, and check if any of its neighbours in $G(N, r)$, had been infected. For this checking, we would select each neighbour of $infec$, at a time, that we defined as *to_check*, and, firstly, make sure it is a susceptible node, through an if condition (i.e. `if infec not in I()`, there is no need to also check if it is not recovered, since $G(N, r)$ has no recovered nodes); and secondly, also check that this node had not been yet checked this iteration (i.e. that it was not part of a list *chekados*, that keeps track of the already checked susceptible nodes this iteration). The reason we make sure it is a susceptible, is to save computational time not checking if infected nodes would be infected again. The reason we check if the node is not in *chekados*, is firstly to save computational time, but secondly and most importantly to not give the opportunity to the susceptible nodes, of becoming infected more than once in the same iteration (we will give more detail about the importance of this condition later). Having selected a susceptible, that is neighbour of at least one infected node, *to_check*, we would use a for loop to go through its edges. Everytime that an edge connected *to_check* with an infected node, we would generate a random sample, r_1, to see if the contact that had been made by them was successful or not (i.e. if `r1 < β`, the contact had been successful). After all the edges of *to_check*, had been checked, if the number of successful contacts would be greater or equal to k, the node *to_check* would be added to $I.temp()$. Since we are checking all the susceptible neighbours

of each infected node, without repeating any, we simulate all the possible dynamics in the network, this iteration.

After all the infected nodes had been selected as $infec$ (i.e. the loop over the nodes of $I()$ had ended), we would attribute the nodes of $I.temp()$ (that includes all the infected nodes prior to this iteration, plus the ones that got infected this iteration) to $I()$. The reason we cannot attribute immediately a susceptible node that just made k (or more) successful contacts, to $I()$, is because this will interfere with the order in which the nodes of $I()$ are checked. As such, as we are going through the infected nodes of $I()$, we could risk selecting as $infec$ a newly infected node (a node that was infected in this iteration), that should only be selected in the next iteration.

Going back to the importance of the array *chekados*, had we not excluded the possibility of susceptible nodes of being checked twice (i.e. to be attributed to the variable *to_check* more than once), we would be passing some nodes through the infection process (i.e. the loop in which we selected each edge of *to_check* to see if any successful contact had been made through it) more than once. If a node of this sort was not infected in a first attempt, it could be infected in a second, so nodes in this situation would have a higher probability of getting infected each time, than nodes that are not in this situation; a discrimination that is not wanted in our model. In fact, a node with N_i infected neighbours, would be given the opportunity to be infected N_i times, and not just one as is proposed by our model. This node would have a probability of not being infected (assuming $N_i \geq k$) of $\left(1 - \sum_{l=k}^{N_i} \binom{N_i}{l} \beta^l (1-\beta)^{N_i - l}\right)^{N_i}$, which is clearly smaller than the wanted probability of $\left(1 - \sum_{l=k}^{N_i} \binom{N_i}{l} \beta^l (1-\beta)^{N_i - l}\right)$.

After there were no more infected nodes, the dynamics of the network would be over, with $G(N, r)$, containing all the remaining susceptible nodes, with the correspondent edges; $I()$ and $I.temp()$ being empty; $R()$ containing all the recovered nodes; and *Recovered_Final()* would represent the original network.

In order to obtain the size of the Biggest Connected Component made entirely of recovered nodes, the RBCC, we cannot use solely $R()$, as this graph only kept the recovered nodes, not their edges. In order to solve this, we removed the nodes that were not recovered (`not in R()`) from *Recovered_Final()*, thus obtaining a graph containing solely the recovered nodes, and their connections in the original graph. In other words, if we could see the spreading of the contagion agent in the network, *Removed Final()* is the graph containing all the recovered nodes, as well as their connections to other recovered nodes, whether they where used in passing the infection of not. Figure 4.1 shows the state of the Original Network in the infinite time limit, as well as the state of the graph *Recovered_Final()*.

Having obtained this version of *Recovered_Final()*, the function *max(nx.connected_components(Recovered_Final()))* provided by the networkx library, was used

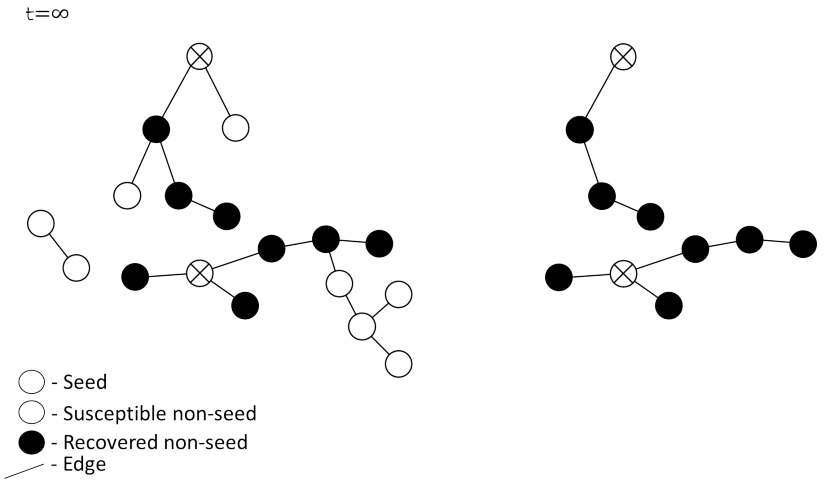


FIGURE 4.1: The state of the Original Network and of *Recovered_Final()* in the infinite time limit. The Original network is displayed on the left, and *Recovered_Final()* is on the right. The empty circles represent the susceptible nodes, the circles with a cross a seed, and the black circles the non-seed nodes that goot infected during the lifetime of the disease

to obtain the biggest connected component of it, the RBCC. By measuring the number of nodes contained in the RBCC, and dividing by N we could finally obtain the fraction of recovered nodes that constituted the RBCC.

We focused on the study of the RBCC as it is the finite system representation of the RGCC. Numerically, one can never simulate a system of infinite size, so an infinite component of recovered nodes, RGCC, can never be obtained (the description and importance of the RGCC are presented in chapter, 2). At most, one can increase N, until reaching a regime where the theory for the infinite size system is applicable (this regime was obtained roughly after $N \geq 1000$). In this regime, one can measure the biggest Recovered component, as is the RBCC, and compare it to the predicted RGCC.

After doing this for a fixed λ, we would increase it by $\delta\lambda$, and repeat all this procedure again, until we reached the condition $S_{RBCC} > 0.95$ (With S_{RBCC} being the fraction of recovered nodes that belonged to the RBCC). We chose this condition since at this point all the critical phenomena had already occurred, in the numerical interval simulated. Another condition to stop λ from increasing was $\lambda > 40$, a condition imposed as firstly, we wanted to make sure that $N \gg \lambda$ (an assumption made for the degree distribution to meet a Poisson one, and for the finite loops to disappear; we state that most our simulations were done with $N = 10,000$), and secondly because these values of λ would take a lot of time to simulate. With all this we obtained the evolution of the RBCC with λ.

A final note should be made that in specific situations, we introduced the variable N_{times}, corresponding to the number of times we would repeat each curve. In these cases, S_{RBCC} would be obtained by averaging out the N_{times} curves. This was done in specific cases in which we wanted to diminish numerical fluctuations.

4.2 Numerical Methods for the Calculation of the Critical Points

As we have seen in the Bootstrap Percolation Model, two types of phase transitions can appear. A continuous, at λ_{crit1}, defining the appearance of the RGCC; and a hybrid one, at λ_{crit2}, representing a discontinuous jump in the evolution of the RGCC. These types of transition also exist in the SIR-Bootstrap Model.

Whilst in the infinite size limit, λ_{crit1} and λ_{crit2} are well defined, in a finite system due to the finite size effect and numerical fluctuations (whose effects we have already explored) these points are harder to measure. Especially since, other than in the SIR and Bootstrap limits, we did not have a theory to guide us where these points should be.

In order to try to determine these critical points in each curve, the following two methods were developed.

4.2.1 Numerical Detection of λ_{crit1}

The RGCC first appears continuously from zero describing a second order phase transition. So, there is a regime when the RGCC is so small, it grows linearly with λ. This is how we theoretically calculated it in the first place, for the SIR and Bootstrap Models. Using this characteristic, the algorithm for detecting λ_{crit1} is based on identifying a linear region after the RGCC starts to grow, in which we can implement a linear regression, whose intersection with the horizontal axis would give us λ_{crit1}. Let us acknowledge that we are trying to calculate the critical point of the RGCC using the RBCC.

The algorithm worked in the following way: We would select a point, λ_0, that looked to be in the linear region of growth of the RGCC. Then, we would select two other values of λ, λ_{max} and λ_{min}, that would define an interval in which the linear region started and ended. λ_{max} would be selected when the curve seemed to be out of the linear region, and λ_{min} would be selected before the beginning of the linear region (usually close to zero, in a near horizontal line that is defined by the RBCC). Figure 4.2 presents an example of the choice of these points.

Next, we would select all the sub-intervals of the points in this interval, that contained λ_0, and perform a linear regression in each. For the linear regression we used the method of least squares provided by the function $stats.linregress()$ of the scipy library. An example of how this worked is the following, say $\lambda_{min} = 3.5$, $\lambda_0 = 4$ and $\lambda_{max} = 5$, with the curve having points every $\delta\lambda = 0.1$. We would calculate a linear regression firstly from using the interval $\lambda \in [3.5, 5]$, then another linear regression from $\lambda \in [3.5, 4.9]$, then $\lambda \in [3.5, 4.8]$,... $\lambda \in [3.6, 5]$,... Until all the combinations that included λ_0, were recorded. The value of the variance of each regression would be kept, and after the process was finished, the regression with the least variance, that had at least N_{points} points in its fit, would be considered the

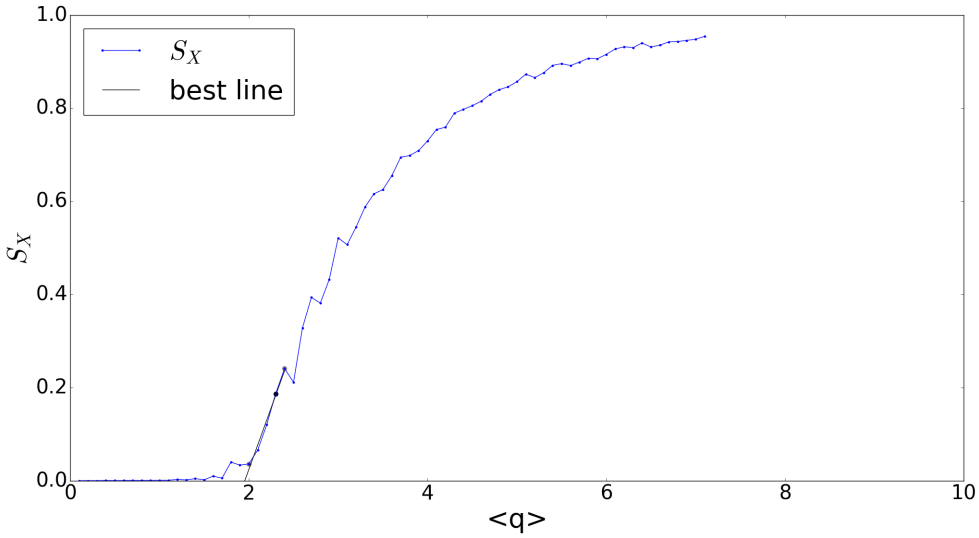


FIGURE 4.2: The evolution of S_{RGCC} with the mean degree for the parameters: $k = 1$, $f = 0.01$, $p = 0.45$, $\alpha = 0.5$. It also presents the best line for the parameters: $\lambda_0 = 2.3$, represented by the black dot; $\lambda_{max} = 4$ and $\lambda_{min} = 2$; $N_{points} = 5$; in gray are presented the points the algorithm chose to define the limits of the interval where the best line would be calculated at.

"best line". Note that by choosing the line with the least variance, we are obtaining the best compromise between including points (that decreases the variance) and decreasing the fluctuations to the linear growth. Finally, we would calculate the intersection of the "best line" with the horizontal axis (the λ axis), obtaining the estimate of the critical point λ_{crit1}.

We required the "best line" to have at least N_{points} points, as had we not imposed this criteria, the chosen line would always have only two points, since the variance of the linear regression using two points is zero. Furthermore, we needed to be careful when choosing N_{points}, so to not allow our algorithm to choose a set of points that due to random fluctuations were disposed linearly, even though they would not represent the linear appearance of the RBCC. For the sake of clarity, Figure 4.3 shows a case where the choice of a too small N_{points}, would predict a wrong λ_{crit1}.

As we can see from Figure 4.3, the points in the interval $\lambda \in [4.2, 4.4]$ form a very good straight line, and are in the middle of the linear region, however, they do not represent the linear growth of the RBCC. On the other hand, one could not set N_{points} too big, to not force our algorithm to include points that are out of the linear growth region of RBCC. Figure 4.4 shows such a case.

Moreover, λ_{crit1} is a point of particularly big numerical fluctuations, so N_{points} had to be carefully chosen as a compromise between including more points to minimize fluctuations, but not so many has to be considering points out of the linear region. Because of this complexity, in addition to the choice of the interval $[\lambda_{min}, \lambda_{max}]$, we did not manage to automate our algorithm (i.e. to set a general way of choosing λ_0, λ_{min} λ_{max} and N_{points}, that would consistently

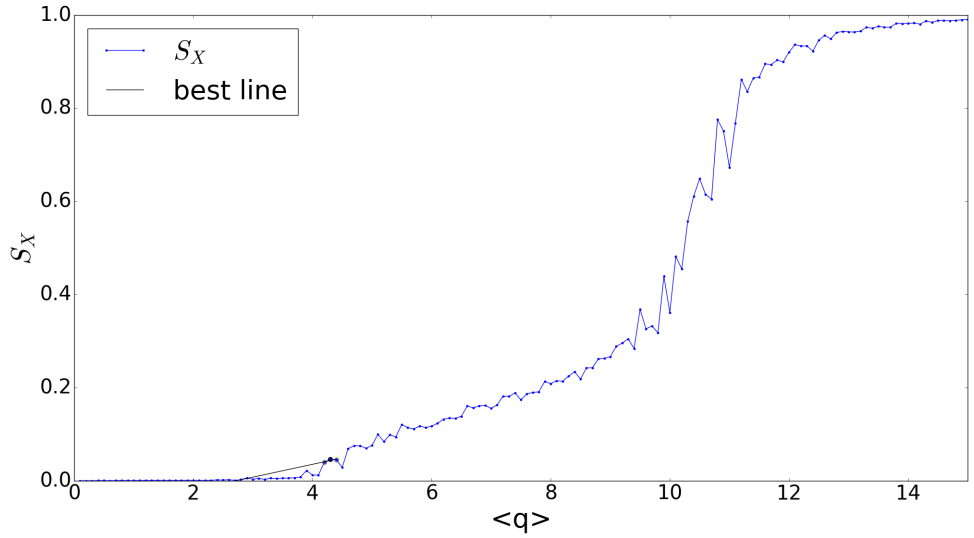


FIGURE 4.3: The evolution of S_{RGCC} with the mean degree for the parameters: $k = 3$, $f = 0.15$, $\beta = 0.9$, $\alpha = 0.27$. It also presents the best fit for the parameters: $\lambda_0 = 4.3$; $\lambda_{max} = 5.7$ and $\lambda_{min} = 3.8$; $N_{points} = 3$; the color coding is the same as the previous Figure 4.2

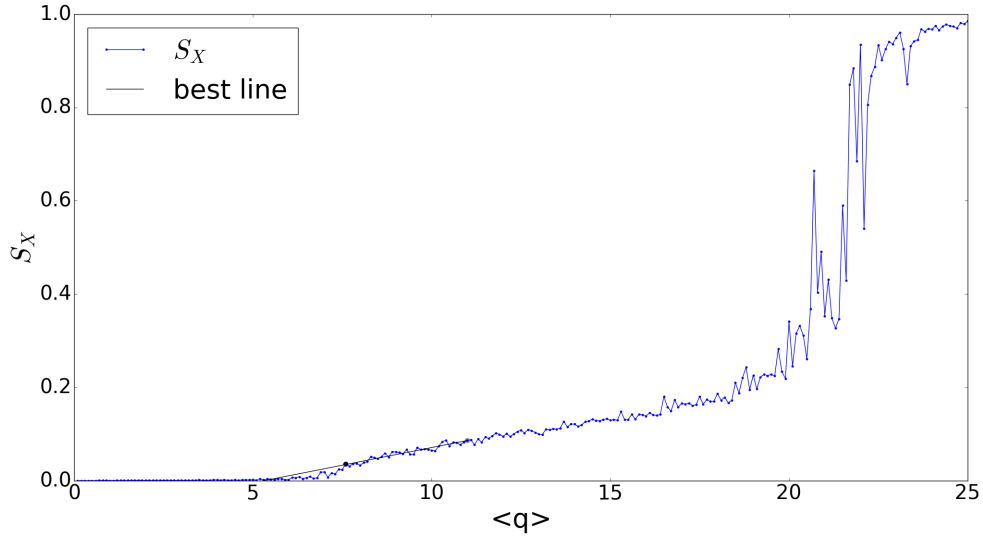


FIGURE 4.4: The evolution of S_{RGCC} with the mean degree for the parameters: $k = 3$, $f = 0.1$, $\beta = 0.9$, $\alpha = 0.5$. It also presents the best fit for the parameters: $\lambda_0 = 7.6$; $\lambda_{max} = 11$ and $\lambda_{min} = 6.7$; $N_{points} = 34$; the color coding is the same as before.

find λ_{crit1}). Figure 4.5 presents two cases in which for a good prediction of λ_{crit1}, required a completely (apparently uncorrelated) choice of parameters.

Also, as explained in the G-GCC, section 2.4, tends to calculate a critical value slightly smaller than the actual one, especially in curves of small linear growth of the RBCC, as the numeric size effect and the numerical fluctuations are of the size of the RBCC in this region, "camouflaging" the critical point. Figure 4.6 shows such a case.

Lastly, Figure 4.7 shows the evolution of the variance with n, the number of points included in each fit, for a given choice of λ_0, λ_{min} and λ_{max}, where we can observe the compromise in

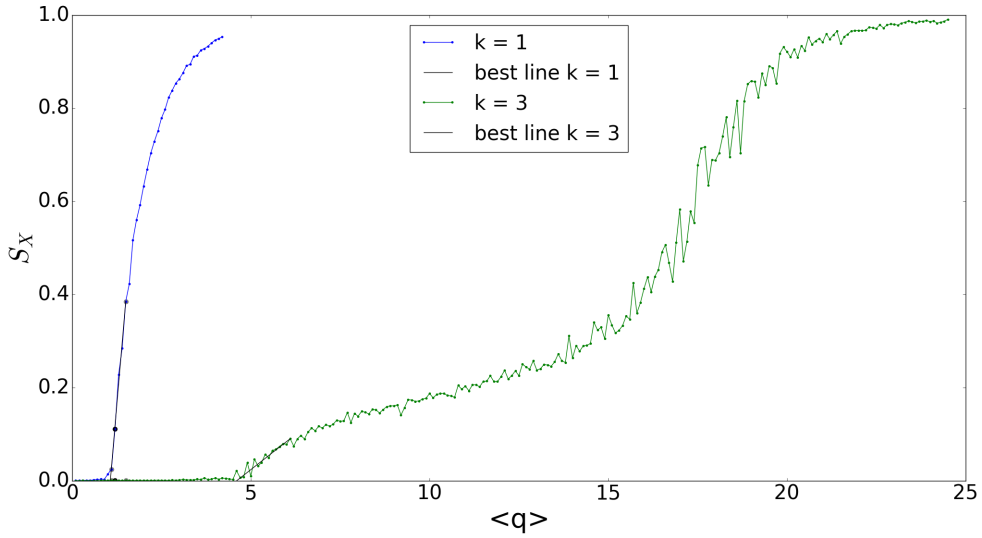


FIGURE 4.5: The evolution of S_{RGCC} with the mean degree. The blue line had parameters $k = 1$, $f = 0.1$, $p = 0.75$, $\alpha = 0.1$, $\lambda_0 = 1.2$; $\lambda_{max} = 1.7$ and $\lambda_{min} = 1$; $N_{points} = 4$. The green line had parameters $k = 3$, $f = 0.16$, $\beta = 0.9$, $\alpha = 0.5$, $\lambda_0 = 5.2$; $\lambda_{max} = 6.7$ and $\lambda_{min} = 4$; $N_{points} = 5$; the color coding is the same as before.

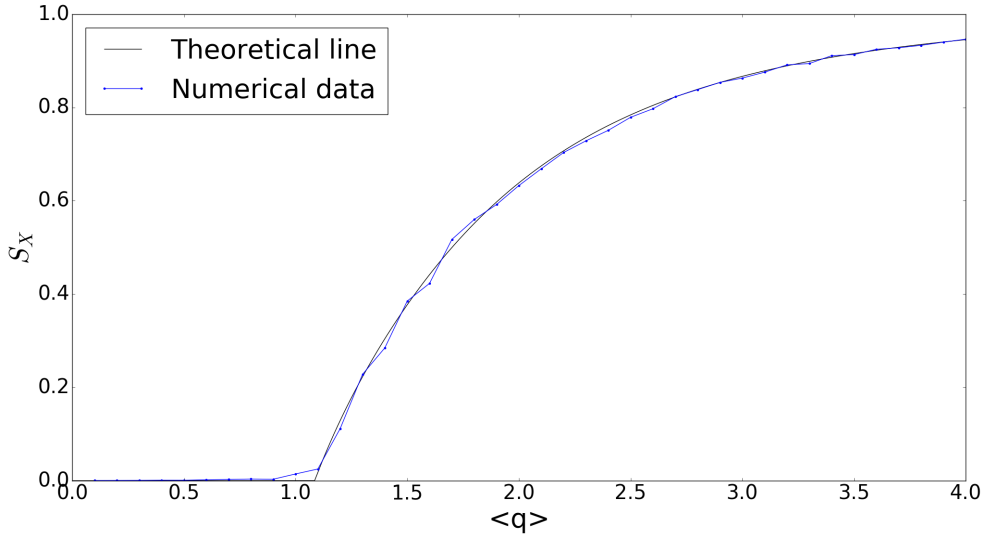


FIGURE 4.6: The evolution of S_{RGCC} with the mean degree for the parameters: $k = 1$, $f = 0.1$, $p = 0.75$, $\alpha = 0.1$. The blue line represents the numerical data, and the black line the theory prediction of the size of the G-GCC.

the choice of N_{points}, to ensure we converge to the correct minimum, not too small nor too large where we would be outside of the linear region. Furthermore, due to fluctuations the value of the wanted minimum was not always as clear as in this figure.

Because of this it is impossible to determine (without a theory) which points are actually the ones that define the linear growth of the RGCC and that should be included in the linear regression, and which ones should not. The inclusion of points that are a result of the finite size effect and of fluctuations, tends to decrease the slope of the "best line", resulting in a

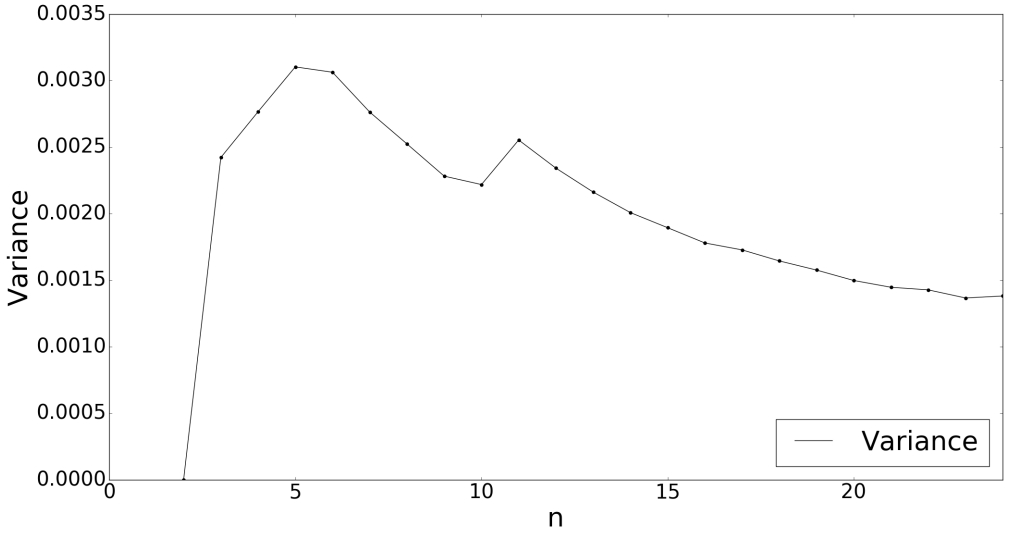


FIGURE 4.7: The evolution of the variance of the data to the correspondent fit, with n. The parameters of the curve were: $k = 3$, $f = 0.16$, $\beta = 0.9$, $\alpha = 0.4$. The first point included in the fit was $\lambda = 4.7$, to which it was added other points, in steps of $\delta\lambda = 0.1$, as we increased n.

smaller λ_{crit1}.

4.2.2 Numerical Detection of λ_{crit2}

In order to construct an algorithm for the determination of λ_{crit2}, notice that the hybrid phase transition shows a discontinuous jump in the solution of $S_{X_{RGCC}}$. However, since we are dealing with finite systems, the point in which this jump happens, λ_{crit2}, is not well defined; furthermore the slope at λ_{crit2} is never divergent (in fact, if we use a too small N, we will not even observe critical phenomena). To counter both these effects, we used bigger networks, diminishing fluctuations, and making the slope at λ_{crit2} steeper, and also used shorter steps of λ, the x-axis. The shorter steps allowed us to diminish the interval of uncertainty of λ_{crit2}, and consequently increase the inclination at λ_{crit2}. Figure 4.8 shows two curves for different values of N and $\delta\lambda$.

Having realized that the slope of $S_{X_{RGCC}}$ is at its steepest at λ_{crit2}, for every set of data, we would use a for loop to go through every interval of 5 points (that usually represented an interval in the mean degree of $\Delta\lambda = 0.5$), and calculate a linear regression through them (using the same scipy function as before). So, the first interval would be $\lambda \in [0, 0.4]$; the second $\lambda \in [0.1, 0.5]$, and so on. After this, we would consider that λ_{crit2} would be in the interval of points that provided the linear regression with the biggest slope. Lastly, we would measure all the differences for each two consecutive points of $S_{X_{RGCC}}$ in this interval, and determine that λ_{crit2} would be the point with the lowest value of λ, of the two points with the biggest difference.

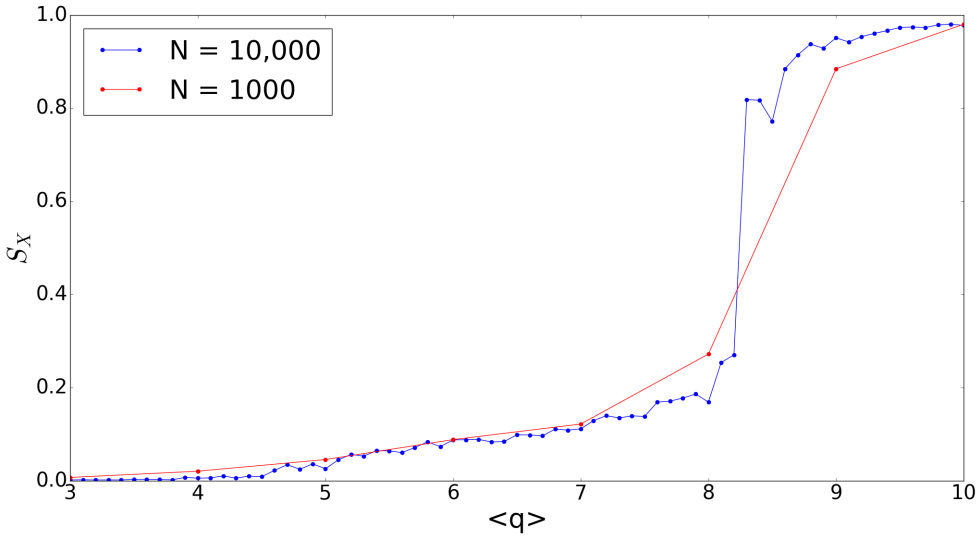


FIGURE 4.8: The evolution of the S_X with $\langle q \rangle$ for different network sizes.

A couple of remarks should be made about this algorithm:

- Firstly, the choice of defining λ_{crit2} as the lowest point between the two that had the greatest difference within the interval of the biggest slope, was just as valid as defining λ_{crit2} as the point with biggest value of λ in these conditions. The choice of one over the other was arbitrary, but had to be made and kept for a matter of consistency.

-Secondly, the choice of considering the size of the interval in which we would do a linear regression to have five points was made upon the observation of a wide range of curves with completely different parameters. This choice would give consistent and very good results to all the different curves. Furthermore, having a regression over five points, gave particularly good results in the simulations that had an step of $\delta\lambda = 0.1$ (which was used for most of our simulations), but also consistent results in curves with bigger steps (although the error could increase).

-Thirdly, the choice of considering the interval of biggest slope to include the critical point, against just choosing λ_{crit2} as the point of biggest jump of the whole curve, was made to give more robustness and consistency to our algorithm. Had we chosen the biggest jump between two consecutive points as λ_{crit2}, we would be more vulnerable to numeric fluctuations, since every now and again, curves could have big jumps that were outside the critical region of λ_{crit2}. This problem was bigger, when the curves had smaller jumps, close to the end of the hybrid phase transitions.

-Fourthly, other similar algorithms, were considered and tested, such as: considering the middle point of the linear regression with the biggest slope as λ_{crit2}; calculating the variance of the points in each interval, instead of calculating the slope, and attributing λ_{crit2} as the middle point in the interval of the biggest variance; calculating the variance as before, but

determining λ_{crit2} to be the smallest of the two consecutive points that had the biggest difference (in $S_{X_{RGCC}}$) between them, in the interval of biggest variance. All these algorithms had good consistency, and provided very similar results. In fact, a sample varying α in a (near) linear region, was used to test and compare these algorithms, that showed that all had a mean standard deviation smaller than $\sigma_{\lambda_{crit2}} < 0.5$ (the mean standard deviation was calculated in relation to the linear regression made with the values that each method gave, in this (apparently) linear regime). The choice in using the algorithm we first described, was made as it was the algorithm that showed the least $\sigma_{\lambda_{crit2}}$, however, by a very short margin. So, had we chosen any of the other algorithms, very similar and just as consistent results would have been obtained.

-Fifthly, notice that the region around the hybrid critical region is very sensitive: first of all, in simulations there are fluctuations in the distribution of edges and the seeds, making the jump (almost) never happen exactly at λ_{crit2}, sometimes it would appear for values a little bigger, sometimes for values a little smaller; furthermore, also due to fluctuations, the size of the jump obtained was rarely of the size of the jump predicted by the theory, in fact, some curves had two (or more) consecutive jumps of half the size that was expected, making the curve look like a "staircase"; in addition, the existence of finite size effects, made the jump not exactly vertical, providing only an interval to search for λ_{crit2}, and increasing the other effects. All these effects made it more difficult to obtain the estimates for λ_{crit2}, especially when the size of the jump was smaller, in spite all this, the way this algorithm was constructed was so consistent that it allowed us to use it in all curves, for any parameter α, β and f, without any further input.

Figure 4.9 shows some curves of very different parameters, α, β and f, where is observable the consistency and efficiency of this algorithm.

4.2.3 Numerical Detection of End of Hybrid Phase Transition

As we have seen in the Bootstrap model, as we increase f for a fixed $\langle q \rangle$ the hybrid phase transition grows smaller until it disappears at f_{crit}. With the introduction of α and β the possibility that the hybrid transition changes behaviour, exists. Since we did not have a theory to predict when this transition ended, we developed a numeric method to try to predict if each curve had or not a hybrid phase transition.

We concluded that the existence of a the hybrid phase transition should be determined by the existence of a jump. If the curve we were analyzing had sufficiently steep change in $S_{X_{RGCC}}$, we would say it had a hybrid phase transition, of critical point λ_{crit2} calculated using the previously presented algorithm. Had the curve not a sufficiently clear change in $S_{X_{RGCC}}$,

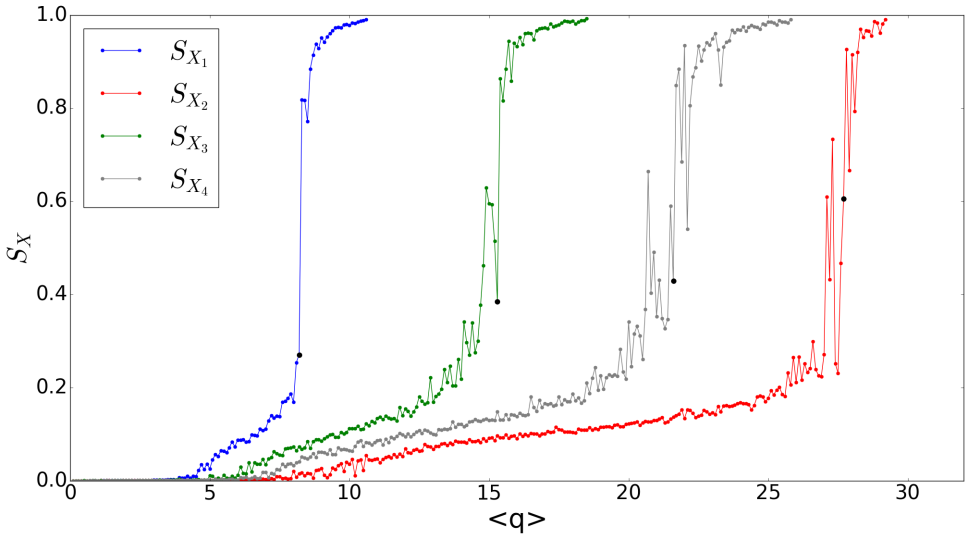


FIGURE 4.9: The value of λ_{crit2}, represented by the black dot, for networks of very different parameters. S_{X_1}: $k = 3$, $f = 0.1$, $\beta = 0.9$, $\alpha = 0.1$; S_{X_2}: $k = 3$, $f = 0.1$, $\beta = 0.2$, $\alpha = 0.1$; S_{X_3}: $k = 3$, $f = 0.1$, $\beta = 0.9$, $\alpha = 0.35$; S_{X_4}: $k = 3$, $f = 0.1$, $\beta = 0.9$, $\alpha = 0.5$.

we would say it did not have a hybrid phase transition. We will define as "leaps" as the differences of $S_{X_{RGCC}}$, for each consecutive value of λ, in our simulations.

To identify if a curve had a jump, we recurred to the curves in which we were certain that (in theory) had a hybrid transition, i.e. the curves in the Bootstrap limit. We studied how these curves evolved with f, especially near the end of the hybrid phase transition. Having calculated the biggest leap of each curve, we obtained Figure 4.10. We define the biggest leap (or maximum leap), has the biggest change in $S_{X_{RBCC}}$, between two subsequent values of λ, in a simulation of a curve.

As expected, the biggest leap diminishes with f, and there is a change of its behaviour around $f = f_{crit}$. Assuming the end of each hybrid transition would behave like the end of the Bootstrap Percolation hybrid transition, we used the curves near f_{crit} to calibrate two thresholds ϵ_1 and ϵ_2.The first, ϵ_1 was defined based on the curves that most certainly had a jump; ϵ_2, was defined based on the curves that we did not consider to have a jump. As such, all the curves whose biggest leap would be bigger than ϵ_1, the upper threshold, were considered to have a phase transition. All the curves whose biggest leap was smaller than ϵ_2, the lower threshold, were considered to not have a phase transition. The curves whose biggest leap was between ϵ_1 and ϵ_2, we would attribute them to an "uncertainty region", marked with smaller points in the Figures, that consisted in a region in which we could not say with certainty if each curve had a jump or not.

An important note should be made about this method: the thresholds were calibrated using the Bootstrap limit, where $\alpha = 0$, and the value of β is indifferent. However, as we increase these parameters, the jump of each curve could change near the end of the hybrid phase

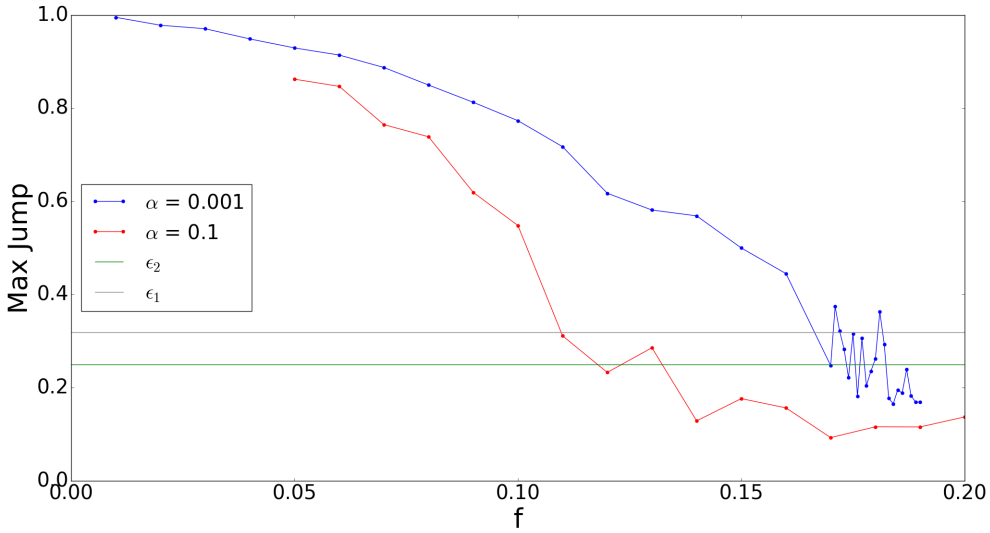


FIGURE 4.10: The value of the maximum leap of each curve with f, for $\alpha = 0.001$ (the Boostrap Limit) and $\alpha = 0.1$, with $k = 3$ and $\beta = 0.9$. The values of the thresholds ϵ_1 and ϵ_2. It is observed the uncertainty region, constituted by the points whose maximum leap is between ϵ_1 and ϵ_2. The value of ϵ_1 was calculated using the maximum leap of the curves: $f = 0.17; 0.173; 0.174; 0.178; 0.179; 0.18; 0.182$, and ϵ_1 of :$f = 0.172; 0.175$, both for $\alpha = 0.001$.

transition. In Figure 4.10 this would correspond to a vertical translation of the line, as is shown in the case $\alpha = 0.1$ and $\beta = 0.9$. However, we tested this method for different types of parameters, and the used values of ϵ_1 and ϵ_2 seemed to give reasonable and consistent results. Nevertheless, one should keep in mind that this method has some uncertainty, hard to quantify, associated with it, and ϵ_1, ϵ_2 and the "uncertainty region" should be looked at qualitatively as "the hybrid transition ends around this region", and not quantitatively as in "the hybrid transition definitively ends within the uncertainty region, and one can say for sure outside of it, when it exists and when it does not". In fact, note that we assume that a hybrid transition exists in general, based on its presence in the Bootstrap limit (when $\alpha \to 0$), however, to numerically prove this we would need to examine the system's finite size scaling effects.

Furthermore, even though these thresholds were calculated using curves around f_{crit}, looked carefully at, compared with curves we knew to have jumps, and with curves we knew to not have jumps, ultimately the criteria to include each curve in the calculation of each threshold was human based, having some subjectivity associated with.

Chapter 5

Results

The SIR-Bootstrap Model has five different parameters essential to study in order to understand the model's behaviour, they are: λ, k, f, α and β. In order to measure the influence of each parameter, and characterize the systems behaviour, we studied how $S_{X_{RGCC}}$ (a Giant Component made solely of recovered nodes) evolved with λ (a curve), having fixed the other four parameters, and calculated λ_{crit1} and λ_{crit2} (when it existed); repeating this for different combinations of the parameters.

Since this is a multi-dimensional problem, we believe the clearest way to present our results is by showing how the critical points evolve with f, α and β, for each value of k. From the critical points, one can understand the general behaviour of each curve, and consequently of the $S_{X_{RGCC}}$ and λ parameters. Furthermore, by treating the cases of k separately we reduce our problem to four dimensions (as we can plot λ_{crit1} and λ_{crit2} in the same axis). Since we could not plot a four dimensional graphic, we decided to cut the four-dimensional space of states in slices, and analyze each slice separately. We define a slice to be the study of the evolution of the critical points with one of the parameters f,α or β, having the other two fixed.

In fact, each figure we present is actually an overlap of a couple of slices. This is because we present different slices in the same figure, that only have in common one fixed parameter. For example, in Figure 5.1 we present the evolution of the critical points with f, for fixed α and β, a slice; however, we show this evolution for different values of α, but always maintaining β. Graphically we could say that each Figure shows the projection of two or three slices in a plane, with only one common fixed parameter. This overlap allows us to compare the influence of two different parameters using only one "overlapped" (two dimensional) slice. This being, the whole system behaviour can be clearly described by three overlapped slices: one for the evolution of the critical points with f, one with α, and another with β.

5.1 Evolution of the Critical Points with f

We will start by analyzing the evolution of the system with f, as this is more familiar to us, from the Bootstrap Model. This is shown in Figure 5.1. Let us remember that β is fixed in the picture, as this will be assumed in the following discussion.

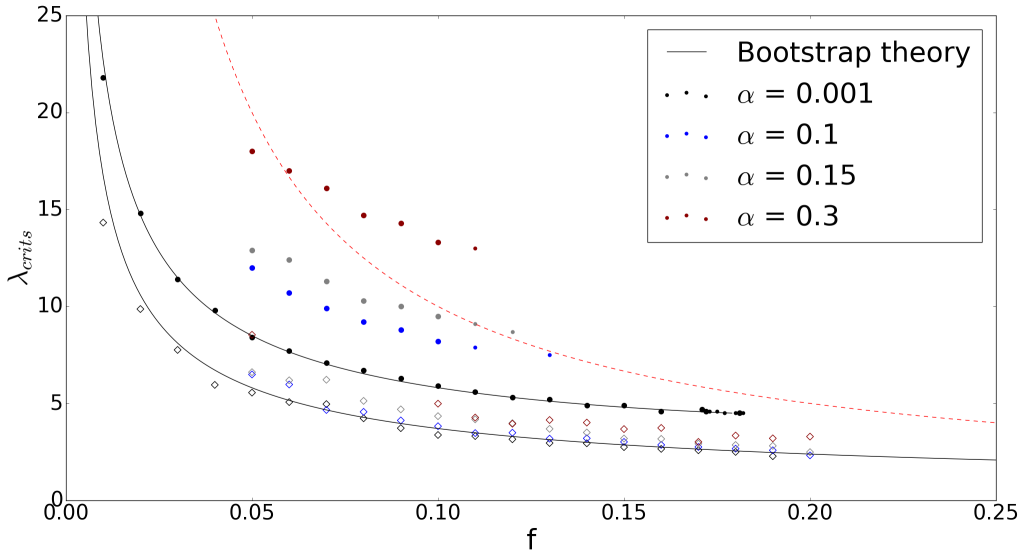


FIGURE 5.1: Evolution of Critical Points with f, for different values of α, $k = 3$ and $\beta = 0.9$. The different values of α are represented by different colors. The values of λ_{crit1} are represented by the hollow diamond shaped dots. The values of λ_{crit2} are represented by the circle dots. The biggest circle dots represent curves whose maximum leap was above ϵ_1, and the smallest circle dots represent curves whose maximum leap was in the "uncertainty region". The upper and lower black lines represent the theory prediction for the Bootstrap limit ($\alpha = 0$), for λ_{crit2} and λ_{crit1}, respectively. The red dashed line represents the upper limit for λ_{crit1}, after which (at least) the seeds percolate into a Giant Component.

We can see that as we increase α, the critical points only appear for higher values of λ. A way to understand this is by comparison with the Bootstrap limit. Let's take the example of a non-seed node that ends up active in the Bootstrap Model in the infinite time limit: if this node is not initially connected to k seeds, it has to wait until some of its neighbours get infected so that they add up to the total of k active neighbours. Since in our model each infected node can recover, it is possible that this node's (less than k) infected neighbours recovered in the time it was waiting for its other neighbours to get infected. In addition, as α increases, this probability increases as well. Furthermore, since $\beta \neq 1$, it is even possible that a node already had k (or more) infected neighbours, but before each of them could make a successful contact in the same time, some of them recovered, making the total number of infected neighbours less than k. The effect of α is compensated by the addition of edges since it allows each node to make contacts with more infected neighbours. So, firstly, if some infected neighbours of a node recovered, it is more likely that the node still has k infected neighbours; and secondly, it is also more likely that at least k of them make a successful contact in each iteration.

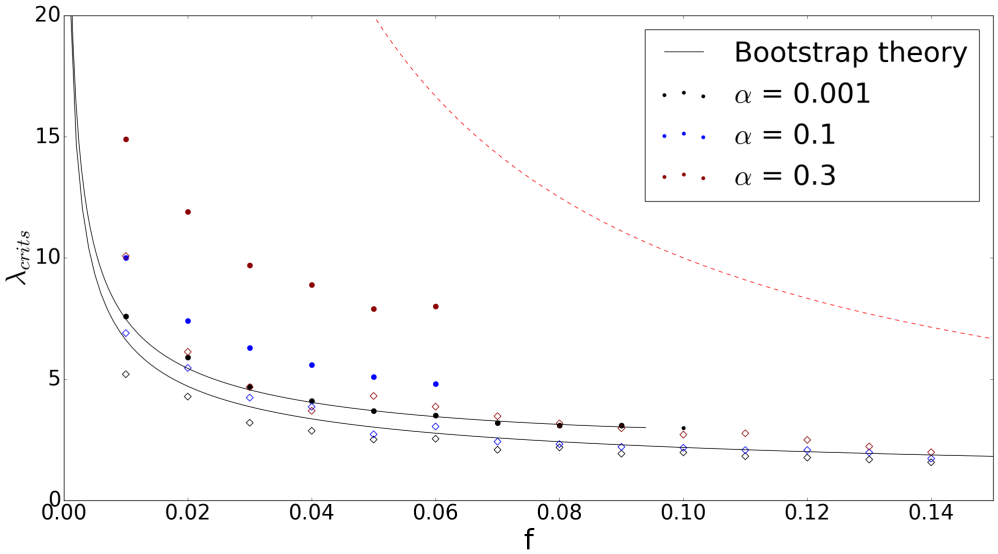


FIGURE 5.2: Evolution of Critical Points with f, for different values of α, $k = 2$ and $\beta = 0.9$. The different values of α are represented by different colors. The values of λ_{crit1} are represented by the hollow diamond shaped dots. The values of λ_{crit2} are represented by the circle dots. The biggest circle dots represent curves whose maximum leap was above ϵ_1, and the smallest circle dots represent curves whose maximum leap was in the "uncertainty region". The upper and lower black lines represent the theory prediction for the Bootstrap limit ($\alpha = 0$), for λ_{crit2} and λ_{crit1}, respectively. The red dashed line represents the upper limit for λ_{crit1}, after which (at least) the seeds percolate into a Giant Component.

All these effects added up, diminish the value size of the Recovered Giant Component for the same λ. The delay of the critical behaviour happens as each recovered component gets smaller as these effects increase (bigger α). In particular for λ_{crit1}, the moment where an infinite recovered component appears (looking at the network in the infinite time limit) is also delayed. The delay of λ_{crit2} can be explained in a similar way. However, a note should be made that we cannot look at the hybrid transition, in this model, as an avalanche in the subcritical clusters of bootstrap percolation, rather we have to look at it as a value of λ that offers a discontinuity in the final number of recovered nodes. This is because the avalanches are a chain event, caused by series of nodes who have exactly $k - 1$ active nodes, that deterministically get active, once one of them becomes active. Because of α and β, the infection (activation) of each node is non-deterministic, so we could have subcritical clusters, that have one of its nodes infected, that does not infect all the other nodes of the subcritical cluster. In terms of dynamics of the network, λ_{crit2} is a point of a sudden increase of the nodes who end up recovered, and we cannot think as in the Bootstrap Percolation Model, since subcritical nodes appear and disapear dynamically. With all this, the existence and increase of α makes it harder for each node to become infected, so the point in which there will be the sudden increase of infected (and then recovered) nodes, λ_{crit2}, should also increase.

An interesting thing to notice is that α has an effect on f_{crit}, the point where the hybrid transition disappears. As we increase α, the hybrid transition disappears sooner (for a smaller

value of f).

Looking at the limits, as expected, the results for $\alpha = 0.001$ behave very similarly to the Bootstrap Percolation theory. In fact, the whole curve of $S_{X_{RGCC}}$ with λ roughly overlapped with the theoretical prediction, meaning that this value of alpha is so small that the infection gets to all the nodes that can be infected, before any of its neighbours recovers. This is enhanced by the fact that Erdős-Rényi graphs have small mean shortest paths. Figure 5.3 presents some pictures of the complete overlap of the curve with the theoretical prediction, for several values of f.

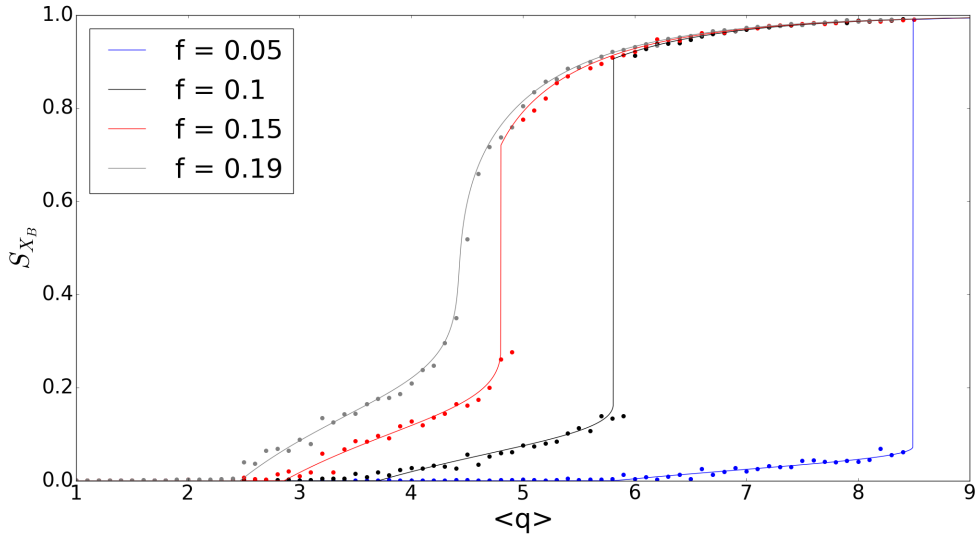


FIGURE 5.3: Evolution of $S_{X_{RGCC}}$ with $\langle q \rangle$ in the Bootstrap limit. The lines represent the different theoretical predictions, and the dots the results, for $k = 3$. The value of α used to simulate the Bootstrap Limit was: $\alpha = 0.001$.

In the limit $f = 1$, all the nodes ends up infected, making the RGCC meet the GCC (all the nodes in the GCC are recovered as well), and also $\lambda_{crit1} = 1$, and of course there cannot be a hybrid transition (it ends somewhen before $f = 1$). In the other limit $f \to 0$, the value for the critical points diverges, as the reduced number of seeds makes it very hard for a node to connect to k of them, let alone to make k successful contacts in the same t, before any of them recovers, nonetheless, for any $f > 0$ there is a λ_{crit1}.

Comparing the figures for $k = 2$ and $k = 3$, Figures 5.1 and 5.2, we can see that the critical points happen sooner for smaller k. This is because, for the same α, β and f, it will always be easier for a node to be infected (and to form infinite recovered components) for smaller values of k. The increase of k offers a restriction in the nodes that can be activated, in other words all the nodes that can be infected in the case for $k = 3$, can also be infected in the case $k = 2$.

5.2 System's Evolution with α

Now we consider the results for the evolution of the critical points with α, shown in Figure 5.5.

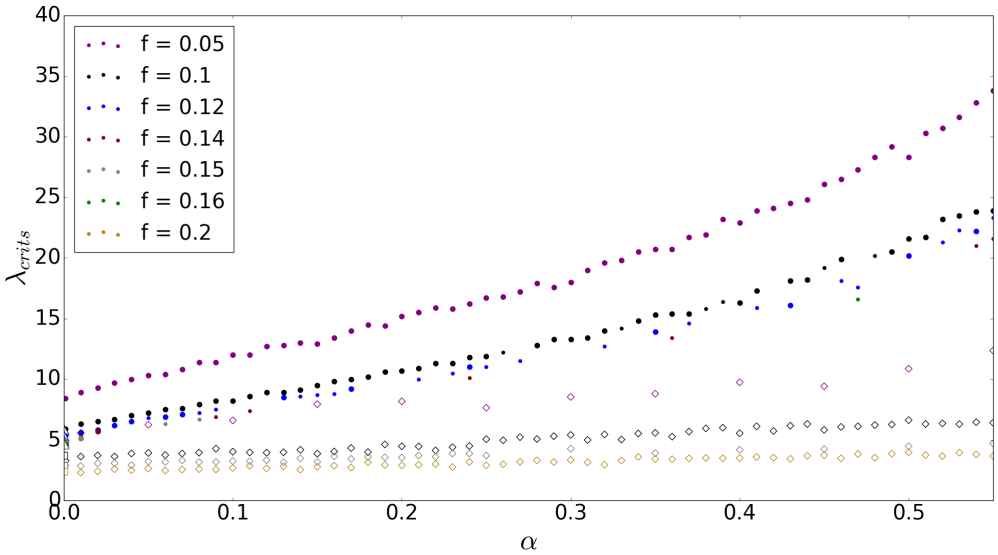


FIGURE 5.4: Evolution of Critical Points with α, for different values of f, $k = 3$ and $\beta = 0.9$. The different values of f are represented by different colors. The values of λ_{crit1} are represented by the hollow diamond shaped dots. The values of λ_{crit2} are represented by the circle dots. The biggest circle dots represent curves whose maximum leap was above ϵ_1, and the smallest circle dots represent curves whose maximum leap was in the "uncertainty region". In the vertical axis (at $\alpha = 0$) are represented the theoretical predictions for the Bootstrap limit, in hollow squares and triangles, for λ_{crit1} and λ_{crit2}, respectively.

The first result we can observe, is once again that, the introduction of the parameter α delays the appearance of the critical behaviour, for the reasons explained before. Furthermore, it seems that the effect of α on λ_{crit2} is much greater than on λ_{crit1}. We can conclude this as the growth of λ_{crit2} with increasing α, is much steeper than the growth of λ_{crit1}. Moreover, it is observed that α has a non-linear effect on λ_{crit2}, as the lines for the smaller seeds (as $f = 0.05$) show some curvature.

The effect of f can be observed in this figure as well. As f increases (for the same λ), the recovered components get bigger since there are more infecting nodes, making the critical phenomena happen sooner.

The evolution of the effect of α on α_{crit} (the α for which this value of f is f_{crit}, i.e. where the hybrid transition ends), changes quickly, resulting in some curves having a hybrid transition at only a few values of α (near $f = f_{crit}$ of the Bootstrap limit), and others, with just a little bigger value of f, having this transition in a broad range of α values. This makes a near horizontal line, defined by the end of each hybrid transition for different values of f. It is also the reason why the uncertainty region is bigger around these values, as small numerical

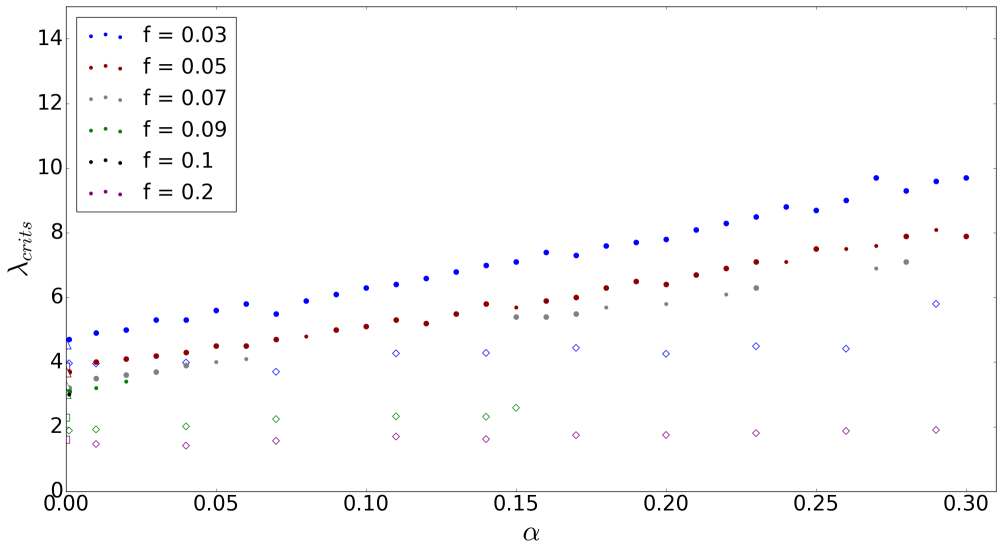


FIGURE 5.5: Evolution of Critical Points with α, for different values of f, $k = 2$ and $\beta = 0.9$. The different values of f are represented by different colors. The values of λ_{crit1} are represented by the hollow diamond shaped dots. The values of λ_{crit2} are represented by the circle dots. The biggest circle dots represent curves whose maximum leap was above ϵ_1, and the smallest circle dots represent curves whose maximum leap was in the "uncertainty region". In the vertical axis (at $\alpha = 0$) are represented the theoretical predictions for the Bootstrap limit, in hollow squares and triangles, for λ_{crit1} and λ_{crit2}, respectively.

fluctuations in the simulation of each curve, would give different results in the algorithm recognizing a jump in the curve or not. This sensitivity in the results is a sign that we should, in fact, be near the end of the hybrid transition, since if there was clearly a jump, or clearly not a jump, the recognition of the jump by the algorithm would be more consistent. The uncertainty region is when the curves show a similar behaviour to the curves that are close to not having a hybrid phase transition, in the Bootstrap limit.

Once again, in the limit of $\alpha \to 0$, the numeric results meet the Bootstrap limit, that in this figure appears only in the vertical axis. In the other limit, $\alpha \to 1$ (and $\beta \neq 1$), we predict both λ_{crit1} and λ_{crit2} to increase. However, λ_{crit1} should meet $\frac{1}{f}$ in this limit, as even though the seeds immediately recover before they can infect anyone, a RGCC should emerge constituted solely by them. This limit is represented by the red dashed line in Figures 5.1 and 5.2.

Once again, and for the same reason as before, we can see that the critical points happen sooner for smaller values of k.

5.3 System's Evolution with β

Lastly, Figure 5.6 presents the overlapped slice that represents the evolution of the system with β.

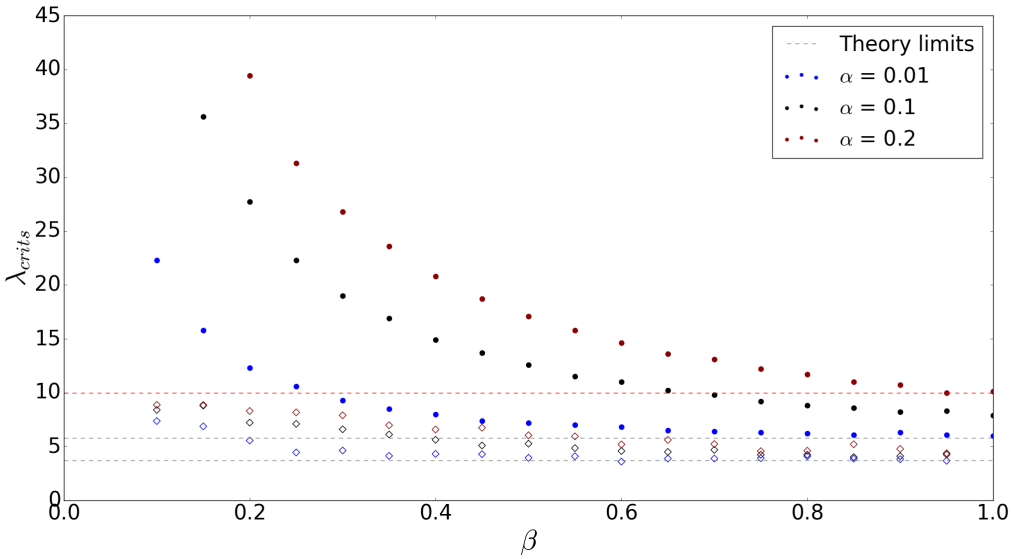


FIGURE 5.6: Evolution of Critical Points with β, for different values of α, $k = 3$ and $f = 0.1$. The different values of α are represented by different colors. The values of λ_{crit1} are represented by the hollow diamond shaped dots. The values of λ_{crit2} are represented by the circle dots. There are no small circle dots, which leads us to believe β does not have an influence on λ_{crit2}. The upper and lower gray dashed lines represent the theory prediction for the Bootstrap limit ($\alpha = 0$), for λ_{crit2} and λ_{crit1}, respectively. The red dashed line represents the upper limit for λ_{crit1}, after which (at least) the seeds percolate into a Giant Component.

The first thing that can be observed is that the increase of β makes the critical behaviour happen sooner. This is because the existence of a $\beta \neq 1$ makes a node with k infected neighbours not become necessarily infected, but only probabilistically. As β decreases, the probability that a susceptible node is exposed to at least k successful contacts by its infected neighbours, in the same timestep, diminishes. Naturally, this makes each recovered cluster smaller, making each phase transition only happen for higher values of mean degree. In a way, the increase of mean degree compensates for smaller values of β, i.e. even though each infected neighbour of a susceptible node has a smaller probability of making a successful contact with it at each timestep, the susceptible node has more (on average) connections to infected neighbours.

Once again, we can observe the delay effect of α on the critical behaviour. As the recovery probability of each infected node increases, each susceptible node is infected with a smaller probability, making the size of the recovered clusters decrease.

The introduction of β also has a non-linear effect on the critical points.

An interesting thing to observe is that β does not have an effect on the presence of the hybrid phase transitions. All configurations that had a hybrid transition before the introduction of the parameter β (i.e. for $\beta \neq 1$), also had this transition no matter what value of β was used (except of course $\beta = 0$). The only difference, as presented before, is that this transition might

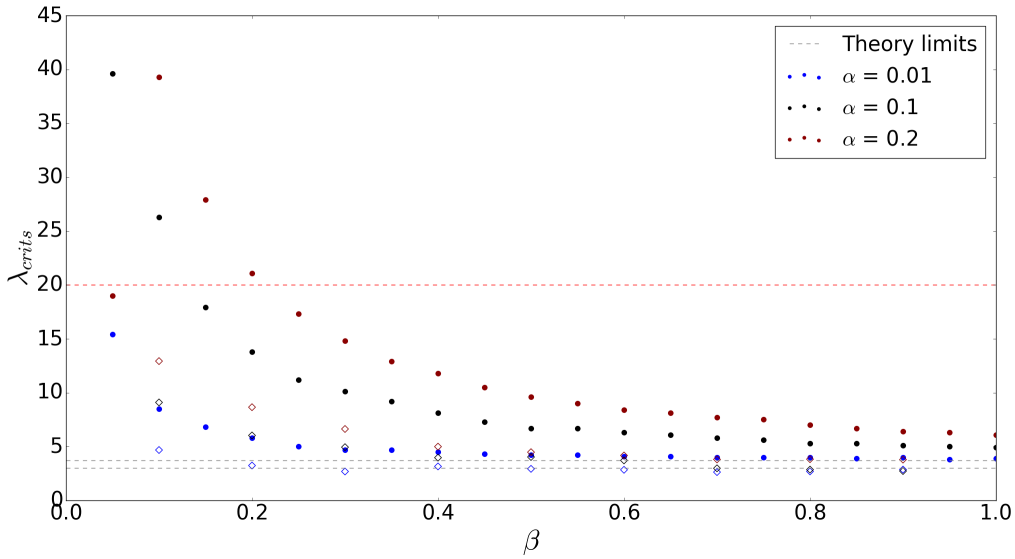


FIGURE 5.7: Evolution of Critical Points with β, for different values of α, $k = 2$ and $f = 0.05$. The different values of α are represented by different colors. The values of λ_{crit1} are represented by the hollow diamond shaped dots. The values of λ_{crit2} are represented by the circle dots. There are no small circle dots, which leads us to believe β does not have an influence on λ_{crit2}. The upper and lower gray dashed lines represent the theory prediction for the Bootstrap limit ($\alpha = 0$), for λ_{crit2} and λ_{crit1}, respectively. The red dashed line represents the upper limit for λ_{crit1}, after which (at least) the seeds percolate into a Giant Component.

happen sooner, or later, and with more fluctuations. Figure 5.8 presents some curves, using different values of β, where this is clearly observed.

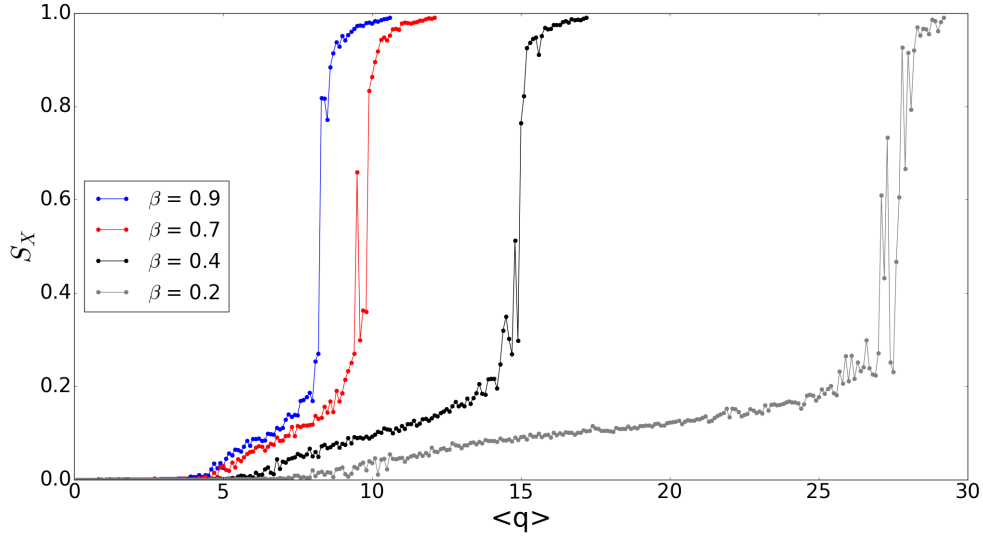


FIGURE 5.8: Evolution of $S_{X_{RGCC}}$ with $\langle q \rangle$, for curves with different values of β. The remaining parameters were: $k = 3$, $f = 0.1$, $\alpha = 0.1$.

We can see that the hybrid transition is delayed, however β does not have an influence on its' existence. In fact, looking at the above figures (and to all our simulations) β might not have

even an influence on the size of the hybrid transition (if this was so, it would be a sufficient condition to justify why it does not disappear).

Also, we would like to point out that the data with smaller values of α are more robust to the decrease of β. In other words, since each infected node recovers with less probability, a susceptible node with at least k infected neighbours, has (on average) more timesteps (iterations) to be infected. This is seen by the fact that the Recovered Giant Component appears sooner for smaller values of α, but also from the fact that, as we decrease β, the slices for smaller α remain "horizontal" during a bigger interval, i.e they behave closer to the Bootstrap limit, where β has no influence. In fact, we predict that the smaller the value of α we use, the longer will the critical points be near the theoretical prediction of the Bootstrap limit. Only growing apart from it for smaller and smaller values of β. In addition, in the case of $\alpha = 0$, we should get points only in the theoretical line for the Bootstrap limit, except at $\beta = 0$.

In the limit $\beta \to 0$, all the critical points increase, as it becomes less likely for a node to be infected. However, for any non-zero value of f (even in the absolute limit $\beta = 0$, where the only recovered nodes are the seeds) there will still be a RGCC, appearing at least at $\lambda = \frac{1}{f}$.

In the case $\alpha = 0$ and $\beta = 0$, the disease does not spread, and the seeds do not recover. Nonetheless, we will observe a Giant Component at $\lambda_{crit1} = \frac{1}{f}$, but made solely of nodes in the Infected state.

A final note should be made, that in the limit $\alpha = 1$ and $\beta = 1$ we have an infinite recovery rate competing with an infinite contagion rate. However, in the way we constructed the model, we check for the recovery of each node first, so the disease will not be able to spread, with the nodes that end up recovered being the seeds, forming a Giant Component at $\frac{1}{f}$.

The points at $\beta = 0.9$ correspond to some of the points in Figure 5.5, in particular, the black points $\beta = 0.9, \alpha = 0.1$ in Figure 5.6, correspond to the also black points at $f = 0.1, \alpha = 0.1$, in Figure 5.5. This allows us to understand the total influence of each parameter: for the same λ, a higher fraction of the network will be recovered for smaller values of α, bigger β, and bigger f, which makes the critical behaviour appear sooner. Also, only the parameters α and f have an influence on the end of the hybrid phase transition. This means that if we were to present another overlapped slice like Figure 5.5, but for another value of β, the curves that had a hybrid phase transition each value of α and f should still have it, only happening for higher values of mean degree.

Lastly, and for the reasons presented before, we would like to highlight that the critical points are smaller for $k = 2$ than for $k = 3$.

Chapter 6

Conclusions

In this thesis was studied the SIR-Bootstrap Model, a generalization of the standard SIR and Bootstrap Percolation Models. The SIR-Bootstrap model holds properties of both these models by giving each infected node a probability β to infect others, and α to recover, meanwhile it requires a threshold, k, of simultaneous successful contacts for a susceptible node to become infected. This model has more applications than its predecessors, as it can be applied to systems in which the number of contacts is an important factor in the spreading of the contagion agent, such as rumors, TV series, as well as introducing a (realistic) probability of infection and recovery, as people do not spread (or watch) the same rumor (series) forever, nor do they assuredly talk about it (and generate interest) with all the people they know.

Numerical methods were developed to simulate the SIR-Bootstrap Model. In these simulations it is observed that the SIR Model and the Bootstrap Percolation Model are limiting cases of the SIR-Bootstrap Model, specifically, when $k = 1$; and when $\alpha = 0$ and $\beta \neq 0$, respectively. As in those models, the SIR-Bootstrap Bootstrap model exhibits a continuous (2nd order) phase transition, beyond which a giant epidemic occurs; and, for $k > 1$, it may also show a discontinuous hybrid phase transition, of the same type as the Bootstrap Percolation Model. Methods for detecting critical points of second order and hybrid phase transitions were also developed, as well as an algorithm to detect the existence of hybrid transitions. All these methods have an error that is hard to quantify, nevertheless they provide solid results when compared to the known theory. This encourages us to believe that the results presented, in the regime where there is no theory, are reliable.

For the case where $k = 1$ we discovered that the classical way of calculating the Recovered Giant Connected Component offers some conceptual ambiguity, and makes an unjustified discrimination in the inclusion/exclusion of some nodes, with the exception of the limit $f \to 0$. As a solution, we created the G-GCC which we believe measures the RGCC in a consistent and correct way. We developed the analytical framework to understand it, noticing that it

meets the CRGCC in the limit $f \to 0$. We believe this explains why the G-GCC has never been noticed (or needed), as usually SIR models are only studied in this limit. Nevertheless, we give examples where infections, and contagion agents start with non-zero seed fraction, highlighting the importance of the G-GCC over the CRGCC. We also studied the critical behaviour of the G-GCC, noticing that a Giant Component appears for smaller values of λ, clearly showing that the discriminated cases of the CRGCC are not negligible. The critical point of the G-GCC (and the evolution of the curve) is fully modulated by the parameters p, a balance between α and β, and f; in contrast with the CRGCC, which shows no dependence on f. We also provide numerical results that support the existence and validity of the G-GCC in the way we described it. Furthermore, we suggested the creation of a Used-GCC (whose mathematical description remains for future work), that would provide additional understanding of the influence of each seed in the spreading of the disease, although not being a sufficient quantity to measure its impact.

For the cases of $k > 1$, we observed that the introduction of a recovery probability, α, delays the appearance of the transitions, increasing the values of λ_{crit1} and λ_{crit2} (the effect being bigger for λ_{crit2}), as well as reducing the size of the RGCC, in comparison with the Bootstrap Percolation Model. We attribute this to the fact that each node has a smaller probability of making at least k successful contacts in a timestep, since some nodes recover (stop being active). Furthermore, the increase of this parameter also anticipates the end of the hybrid phase transition, f_{crit}. All these behaviours were observed by numerical simulations of the SIR-Bootstrap Model.

Referring still to the cases of $k > 1$, the introduction of β also increases the values of λ_{crit1} and λ_{crit2}. We attribute this to the reduced probability of each node making k successful contacts in the same timestep, thus diminishing the size of the recovered clusters, and consequently of the critical behaviour. However, the introduction of this parameter does not have an effect on f_{crit}, as all curves that had a hybrid transition before the introduction of this parameter ($\beta = 1$), continue to have this transition.

In our simulations it is observed that with the increase of the number of nodes in the network, we would obtain better results: results closer to the theory (made assuming $N \to \infty$); with less fluctuations and finite size effect. From $N > 1000$ critical behaviour could start to be observed in a clear way, and was distinct for networks with $N = 10000$ (the value which the results we presented were calculated). Due to limitations in computer power we did not use networks of $N > 10000$. The usage of bigger networks would present less noisy data, improving our results, and in particular reduce the uncertainty around the calculation of the end of the hybrid phase transiton, f_{crit}. Moreover, we could have presented two 3D plots (for each k), in which it would be seen a more clear influence of each parameter.

For future research remains the discovery of a full mathematical description of the SIR-Bootstrap model, other than in the SIR and Bootstrap Percolation limits; which even though was not achieved we believe it to be possible. This study also prompts the application of the SIR-Bootstrap Model in other types of networks, such as networks with divergent second moment of their degree distribution (as Scale-Free Networks). Generalizations could be made where individual values of β, α and k would be attributed to each node; as well as the introduction of other states (compartments); or even the study of the SIR-Bootstrap model in a continuous time frame.

Bibliography

[1] K. Lee, A. Goltsev, M. Lopes and J. Mendes, in *AIP Conference Proceedings*, Vol. 1510 (AIP, 2013) pp. 195–201.

[2] M. E. Newman, *The structure and function of complex networks*, SIAM review **45**, 167 (2003).

[3] D. Witthaut, M. Rohden, X. Zhang, S. Hallerberg and M. Timme, *Critical links and nonlocal rerouting in complex supply networks*, Physical review letters **116**, 138701 (2016).

[4] C. Han, D. Witthaut, M. Timme and M. Schröder, *The winner takes it all—Competitiveness of single nodes in globalized supply networks*, PloS one **14**, e0225346 (2019).

[5] W. H. Organization, "Smallpox," (2019).

[6] H. S. Rodrigues, *Application of SIR epidemiological model: new trends*, arXiv preprint arXiv:1611.02565 (2016).

[7] L. J. Allen, F. Brauer, P. Van den Driessche and J. Wu, *Mathematical epidemiology*, Vol. 1945 (Springer, 2008).

[8] A. Goltsev, F. De Abreu, S. Dorogovtsev and J. Mendes, *Stochastic cellular automata model of neural networks*, Physical Review E **81**, 061921 (2010).

[9] D. Centola and M. Macy, *Complex contagions and the weakness of long ties*, American journal of Sociology **113**, 702 (2007).

[10] W. O. Kermack and A. G. McKendrick, *A contribution to the mathematical theory of epidemics*, Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character **115**, 700 (1927).

[11] S. Moore and J. L. Simon, *The greatest century that ever was: 25 miraculous trends of the past 100 years* (Cato Institute, 1999).

[12] C. for Disease Control and Prevention, "Types of influenza viruses," (2019).

[13] C. for Disease Control and Prevention, "Diseases you almost forgot about (thanks to vaccines)," (2019).

[14] R. Cohen, K. Erez, D. Ben-Avraham and S. Havlin, *Resilience of the internet to random breakdowns*, Physical review letters **85**, 4626 (2000).

[15] N. Madar, T. Kalisky, R. Cohen, D. Ben-avraham and S. Havlin, *Immunization and epidemic dynamics in complex networks*, The European Physical Journal B **38**, 269 (2004).

[16] D. Brockmann and D. Helbing, *The hidden geometry of complex, network-driven contagion phenomena*, science **342**, 1337 (2013).

[17] M. Gardner, *Mathematical games*, Scientific American **222**, 132 (1970).

[18] J. Soriano, M. R. Martínez, T. Tlusty and E. Moses, *Development of input connections in neural cultures*, Proceedings of the National Academy of Sciences **105**, 13758 (2008).

[19] J.-P. Eckmann, O. Feinerman, L. Gruendlinger, E. Moses, *et al.*, *The physics of living neural networks*, Physics Reports **449**, 54 (2007).

[20] M. Sellitto, G. Biroli and C. Toninelli, *Facilitated spin models on Bethe lattice: Bootstrap percolation, mode-coupling transition and glassy dynamics*, EPL (Europhysics Letters) **69**, 496 (2005).

[21] C. Toninelli, G. Biroli and D. S. Fisher, *Jamming percolation and glass transitions in lattice models*, Physical review letters **96**, 035702 (2006).

[22] S. Sabhapandit, D. Dhar and P. Shukla, *Hysteresis in the random-field Ising model and bootstrap percolation*, Physical Review Letters **88**, 197202 (2002).

[23] A. Holroyd, *et al.*, *The metastability threshold for modified bootstrap percolation in d dimensions*, Electronic Journal of Probability **11**, 418 (2006).

[24] G. J. Baxter, S. N. Dorogovtsev, A. V. Goltsev and J. F. Mendes, *Bootstrap percolation on complex networks*, Physical Review E **82**, 011103 (2010).

[25] D. J. Watts, *A simple model of global cascades on random networks*, Proceedings of the National Academy of Sciences **99**, 5766 (2002).

[26] M. A. Di Muro, L. D. Valdez, H. E. Stanley, S. V. Buldyrev and L. A. Braunstein, *Insights into bootstrap percolation: Its equivalence with k-core percolation and the giant component*, Physical Review E **99**, 022311 (2019).

[27] B. Min and M. San Miguel, *Competing contagion processes: Complex contagion triggered by simple contagion*, Scientific reports **8**, 10422 (2018).

[28] P. Erdős and A. Rényi, *On the evolution of random graphs*, Publ. Math. Inst. Hung. Acad. Sci **5**, 17 (1960).

[29] S. Dorogovtsev, *Lectures on Complex Networks* (Oxford University Press, Inc., USA, 2010).

[30] G. Bianconi and M. Marsili, *Loops of any size and Hamilton cycles in random scale-free networks*, Journal of Statistical Mechanics: Theory and Experiment **2005**, P06005 (2005).

[31] L. K. Gallos, R. Cohen, P. Argyrakis, A. Bunde and S. Havlin, *Stability and topology of scale-free networks under attack and defense strategies*, Physical review letters **94**, 188701 (2005).

[32] M. E. Newman, *Spread of epidemic disease on networks*, Physical review E **66**, 016128 (2002).

[33] A. M. Lyapunov, *The general problem of the stability of motion*, International journal of control **55**, 531 (1992).

[34] H. Hong, M. Ha and H. Park, *Finite-size scaling in complex networks*, Physical review letters **98**, 258701 (2007).

[35] S. N. Dorogovtsev, A. V. Goltsev and J. F. Mendes, *Critical phenomena in complex networks*, Reviews of Modern Physics **80**, 1275 (2008).

[36] Python Core Team, "Python," .

[37] Jupyter Team, "Jupyter," .

[38] Travis Oliphant, "Numpy," .

[39] A. Hagberg, P. Swart and D. S Chult, *Exploring network structure, dynamics, and function using NetworkX*, Tech. Rep. (Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008).

[40] J. D. Hunter, *Matplotlib: A 2D graphics environment*, Computing in Science & Engineering **9**, 90 (2007).

[41] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, *et al.*, *SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python*, arXiv e-prints , arXiv:1907.10121 (2019), arXiv:1907.10121 [cs.MS] .

[42] A. V. Goltsev, S. N. Dorogovtsev, J. G. Oliveira and J. F. Mendes, *Localization and spreading of diseases in complex networks*, Physical review letters **109**, 128702 (2012).

[43] M. E. Newman, S. H. Strogatz and D. J. Watts, *Random graphs with arbitrary degree distributions and their applications*, Physical review E **64**, 026118 (2001).

[44] R. Pastor-Satorras, C. Castellano, P. Van Mieghem and A. Vespignani, *Epidemic processes in complex networks*, Reviews of modern physics **87**, 925 (2015).

[45] S. Yoon, A. Goltsev and J. Mendes, *Structural stability of interaction networks against negative external fields*, Physical Review E **97**, 042311 (2018).

[46] J. Balogh and B. Bollobás, *Bootstrap percolation on the hypercube*, Probability Theory and Related Fields **134**, 624 (2006).

[47] J. Balogh and B. G. Pittel, *Bootstrap percolation on the random regular graph*, Random Structures & Algorithms **30**, 257 (2007).